December 1988 UILU-ENG-88-2264

(2)

COORDINATED SCIENCE LABORATORY
College of Engineering

EXPLANATION-BASED THEORY REVISION: AN APPROACH TO THE PROBLEMS OF INCOMPLETE AND INCORRECT THEORIES

Shankar A. Rajamoney

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified	1b. RESTRICTIVE MARKINGS: None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): UILU-ENG-88-2264	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Coordinated Science Lab, University of Illinois; 6b. OFFICE SYMBOL (If applicable): N/A	7a. NAME OF MONITORING ORGANIZATION: Office of Naval Research
6c. ADDRESS (City, State, and ZIP Code): 1101 W. Springfield Ave. Urbana, IL 61801	7b. ADDRESS (City, State, and ZIP Code): Arlington, VA 22217
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Office of Naval Research; 8b. OFFICE SYMBOL (If applicable):	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-86-K-0309
8c. ADDRESS (City, State, and ZIP Code): Arlington, VA 22217	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
EXPLANATION-BASED THEORY REVISION: AN APPROACH TO THE PROBLEMS OF INCOMPLETE AND INCORRECT THEORIES

12. PERSONAL AUTHOR(S)
Rajamoney, Shankar Anandsubramaniam

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM _____ TO _____	December 1988	237

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	imperfect theory problems, theory revision, explanation-based learning, learning qualitative theories,

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Knowledge-intensive Artificial Intelligence systems rely on a model of the domain, called a *domain theory*, to fulfill their tasks. A domain theory consists of an encoding of the knowledge required by the system to draw inferences about situations of interest. Systems that rely on a domain theory face two difficult problems. 1) Their performance is directly related to the amount of knowledge in the domain theory. In order to insure a satisfactory level of performance, the expect who constructs the domain theory has the tedious chore of anticipating the wide variety of examples on which the system may be run. For most complex real-world domains it is impossible to anticipate and and handcode all the required knowledge. The expert is forced to make approximations and assumptions. This results in brittle systems that tend to draw erroneous inferences and fail frequently. 2) Systems that rely on a domain theory are limited to reasoning within the deductive closure of the knowledge in the domain theory. Since the knowledge content of the domain theory remains constant, these systems are incapable of modelling dynamic or under-specified domains in which new knowledge is being constantly acquired. Furthermore, large amounts of additional knowledge must be provided to the system if it is to process new examples. Consequently, such systems tend to be inflexible and inextensible.

over

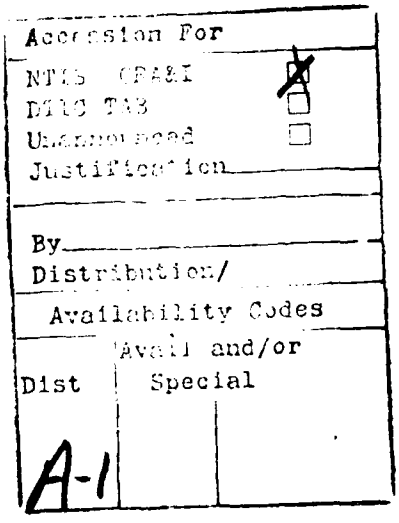
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

UNCLASSIFIED

19. Abstract (continued)

This thesis describes a method called *explanation-based theory revision* for augmenting and correcting an inadequate domain theory. In brief, the method consists of detecting failures due to the inadequacies of the domain theory. In brief, the method consists of detecting failures due to the inadequacies of the domain theory, hypothesizing modifications or additions to the domain theory to eliminate the failures, designing experiments to obtain additional information to refute incorrect hypotheses, recalling previous experiences of the system to reject inconsistent revisions, and selecting a best theory from the remaining theories based on aesthetic criteria. Explanation-based theory revision addresses both the problems described above. The expert use "quick and dirty" methods to build and operational domain theory. Explanation-based theory revision monitors the performance of the system using the domain theory. If failures occur, explanation-based theory revision modifies the domain theory to elimina'e the failures. Furthermore, explanation-based theory revision augments the domain theory by incorporating the additional information obtained through experimentation into the domain theory. Consequently, the system learns at the knowledge level, that is, the deductive closure of the knowledge in the domain theory changes.

Explanation-based theory revision has been incorporated into a system called COAST. COAST revises qualitative theories of the physical world. COAST has been demonstrated on a number of examples involving the revision of qualitative theories about physical phenomena such as evaporation, osmosis, flow of fluids, dissolving of substances, chemical decomposition of compounds, and combustion.



EXPLANATION-BASED THEORY REVISION: AN APPROACH TO THE PROBLEMS OF INCOMPLETE AND INCORRECT THEORIES

BY

SHANKAR ANANDSUBRAMANIAM RAJAMONEY

B.Tech., Indian Institute of Technology, 1983
M.S., University of Illinois, 1986

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1989

Urbana, Illinois

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

JANUARY 1989

WE HEREBY RECOMMEND THAT THE THESIS BY

SHANKAR ANANDSUBRAMANIAM RAJAMONEY

ENTITLED EXPLANATION-BASED THEORY REVISION:

AN APPROACH TO THE PROBLEMS OF INCOMPLETE AND INCORRECT THEORIES

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF DOCTOR OF PHILOSOPHY

Director of Thesis Research

Head of Department

Committee on Final Examination†

Chairperson

Robert E. Stepp

Uday Reddy

† Required for doctor's degree but not for master's.

EXPLANATION-BASED THEORY REVISION:
AN APPROACH TO THE PROBLEMS OF INCOMPLETE AND INCORRECT THEORIES

Shankar Anandsubramaniam Rajamoney, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1989
Gerald Francis DeJong, Advisor

Knowledge-intensive Artificial Intelligence systems rely on a model of the domain, called a *domain theory*, to fulfill their tasks. A domain theory consists of an encoding of the knowledge required by the system to draw inferences about situations of interest. Systems that rely on a domain theory face two difficult problems. 1) Their performance is directly related to the amount of knowledge in the domain theory. In order to insure a satisfactory level of performance, the expert who constructs the domain theory has the tedious chore of anticipating the wide variety of examples on which the system may be run. For most complex real-world domains it is impossible to anticipate and handcode all the required knowledge. The expert is forced to make approximations and assumptions. This results in brittle systems that tend to draw erroneous inferences and fail frequently. 2) Systems that rely on a domain theory are limited to reasoning within the deductive closure of the knowledge in the domain theory. Since the knowledge content of the domain theory remains constant, these systems are incapable of modelling dynamic or under-specified domains in which new knowledge is being constantly acquired. Furthermore, large amounts of additional knowledge must be provided to the system if it is to process new examples. Consequently, such systems tend to be inflexible and inextensible.

This thesis describes a method called *explanation-based theory revision* for augmenting and correcting an inadequate domain theory. In brief, the method consists of detecting failures due to the inadequacies of the domain theory, hypothesizing modifications or additions to the domain theory to eliminate the failures, designing experiments to obtain additional information to refute incorrect hypotheses, recalling previous experiences of the system to reject inconsistent revisions, and selecting a best theory from the remaining theories based on aesthetic criteria. Explanation-based theory revision addresses both the problems described above. The expert uses "quick and dirty" methods to build an operational domain theory. Explanation-based theory revision monitors the performance of the system using the domain theory. If failures occur, explanation-based theory revision modifies the domain theory to eliminate the failures. Furthermore, explanation-based theory revision augments the domain theory by incorporating the additional information obtained through

experimentation into the domain theory. Consequently, the system learns at the knowledge level, that is, the deductive closure of the knowledge in the domain theory changes.

Explanation-based theory revision has been incorporated into a system called COAST. COAST revises qualitative theories of the physical world. COAST has been demonstrated on a number of examples involving the revision of qualitative theories about physical phenomena such as evaporation, osmosis, flow of fluids, dissolving of substances, chemical decomposition of compounds, and combustion.

DEDICATION

To my parents,

Rajalakshmi and V. A. Rajamoney

ACKNOWLEDGMENTS

I thank:

Gerald DeJong, my advisor, for providing a constant source of ideas and inspiration, for numerous thought-provoking and fruitful discussions and for providing a friendly and intellectually stimulating research environment.

Ken Forbus for copious comments on the thesis, for many helpful discussions on qualitative reasoning and learning qualitative theories, and for inventing Qualitative Process theory.

Dedre Gentner, Uday Reddy and Bob Stepp for also serving on my final examination committee, for providing different perspectives on the research, and for many helpful suggestions, criticism and advice on the thesis.

Scott Bennett, Steve Chien, Melinda Gervasio and Larry Watanabe, the members of the explanation-based learning group, for useful discussions on explanation-based learning, imperfect theory problems and theory revision.

Raymond Mooney, Paul O' Rorke, Alberto Segre and Jude Shavlik, the previous members of the explanation-based learning group, for helpful discussions during the early stages of this research.

Brian Falkenhainer for many interesting discussions on analogical learning, theory formation, scientific discovery and qualitative reasoning.

Bob Stepp and his students – Diane Cook, Larry Holder, Bharat Rao, Bob Reinke, and Brad Whitehall – for providing a better understanding of similarity and difference-based learning and for helpful discussions on theory revision and experimentation.

And finally, Maithili for her love and understanding that carried me through this last year.

Support for this research was provided by a University of Illinois Fellowship, four University of Illinois Cognitive Science/Artificial Intelligence Fellowships, and by the Office of Naval Research under grant N-00014-86-K-0309.

TABLE OF CONTENTS

CHAPTER

1 INTRODUCTION 1

1.1. Introduction 1

1.2. Knowledge-Intensive Systems 3

1.3. Overview of Explanation-based Theory Revision 4

1.4. Organization of the Thesis 5

2 OVERVIEW OF EXPLANATION-BASED THEORY REVISION 7

2.1. Explanation-based Theory Revision 7

2.2. COAST 9

2.2.1 Knowledge Representation in COAST 9

2.2.2. Satisfying the Requirements for Explanation-based Theory Revision 15

2.2.3. Examples Implemented in COAST 16

2.3. Examples of COAST's Theory Revision 16

2.4. Summary 24

3 THE CLASSIFICATION AND DETECTION OF IMPERFECT THEORY PROBLEMS 25

3.1. Introduction 25

3.2. The Classification of Imperfect Theory Problems 25

3.3. Examples of Incomplete and Incorrect Domain Theories in QP Theory 27

3.4. The Detection of Imperfect Theory Problems 31

3.5. The Detection of Incomplete and Incorrect Theory Problems in COAST 32

3.5.1. Unexpected Observation Failure 34

3.5.2. Failed Prediction Failure 37

3.5.3. Inverse Behavior Failure 40

3.6. Discussion 43

4 HYPOTHESIS GENERATION FOR THEORY REVISION 44

4.1. Constraints for Hypothesis Generation 44

4.1.1. Theory Revision Operators 44

4.1.2. Scenario Constraint 46
4.1.3. Explanation Constraint 48
4.1.4. Abstraction Constraint 50
4.1.5. Building an Effective Hypothesis Generator 53
4.2. Hypothesis Generation in COAST 54
4.3. Abstract Hypotheses 55
4.4. Explanation Construction based on Abstract Hypotheses 58
4.4.1. Explanation Construction for Unexpected Observation Failures 59
4.4.1.1. Causes? Explanations 59
4.4.1.2. Active? Explanations 61
4.4.1.3. New-Process? Explanations 62
4.4.2. Explanation Construction for Failed Prediction Failures 63
4.4.2.1. Inactive? Explanations 64
4.4.2.2. Not-Causes? Explanations 65
4.4.2.3. Unexpected-Observation? Explanations 66
4.4.3. Explanation Construction for Inverse Behavior Failures 67
4.4.3.1. Causes? Explanations 68
4.4.3.2. Unexpected-Observation? Explanations 69
4.5. Refining Hypotheses in QP Theory 70
4.5.1. Theory Revision Operators for QP Theory 71
4.5.2. Refining Abstract Hypotheses 73
4.5.2.1. Active? 73
4.5.2.2. Causes? 76
4.5.2.3. Inactive? 79
4.5.2.4. Not-causes? 82
4.5.2.5. Equals? 83
4.5.2.6. Dominates? 83
4.5.2.7. Unexpected-observation? 83
4.5.2.8. New-process? 84
4.6. Discussion 85
5 EXPERIMENTATION-BASED HYPOTHESIS REFUTATION 87
5.1. Introduction 87

5.2. Experimentation–based Hypothesis Refutation 90
5.2.1. Strategies for Experiment Design 92
5.3. Experimentation–based Hypothesis Refutation in COAST 95
5.3.1. An Experiment Engine 95
5.3.2. Obtaining Predictions for Theory Revision Hypotheses 97
5.3.3. Strategies for Experiment Design 101
5.3.3.1. Elaboration 101
5.3.3.2. Discrimination 104
5.3.3.3. Transformation 105
5.4. Evaluation of Experimentation–based Hypothesis Refutation 112
5.4.1. Efficacy 112
5.4.2. Efficiency 113
5.4.3. Tolerance of Unavailable Data 113
5.4.4. Feasibility 114
5.5. Discussion 115
6 EXEMPLAR–BASED THEORY REJECTION 116
6.1. Introduction 116
6.2. Exemplar–based Theory Rejection 119
6.2.1. Exemplars and Prototypes 119
6.2.2. Forming the Exemplar Space 121
6.2.3. Using the Exemplar Space 123
6.3. Exemplar–based Theory Rejection in COAST 124
6.3.1. Requirements of an Exemplar Space 125
6.3.2. Forming the Exemplar Space 126
6.3.2.1. Examples 127
6.3.2.2. Evaporation of a Solution of Salt 127
6.3.2.3. Dissolving Salt in a Salt Solution 130
6.3.2.4. Some Additional Examples 133
6.3.3. Using the Exemplar Space 136
6.3.3.1. Examples 139
6.3.3.2. Evaporation of Alcohol 139
6.3.3.3. Evaporation of a Sugar Solution 142

6.4. Evaluation of Exemplar-based Theory Rejection 146
6.4.1. Efficacy 149
6.4.2. Efficiency 150
6.4.3. Oscillation 150
6.5. Discussion 151
7 SELECTION OF THEORIES 153
7.1. Introduction 153
7.2. Theory Selection Criteria 154
7.3. Computing Estimates for the Criteria in COAST 154
7.3.1. The Structural Simplicity of the Theory 154
7.3.2. The Simplicity of the Explanations Constructed by the Theory 158
7.3.3. The Predictive Power of the Theory 161
7.3.4. Combining the Three Criteria 164
7.4. Discussion 165
8 ADDITIONAL APPLICATIONS OF EXPLANATION-BASED THEORY REVISION 166
8.1. Introduction 166
8.2. The Multiple Explanations Problem in Explanation-Based Learning 166
8.2.1. Experimentation-based Hypothesis Refutation and the Multiple Explanations Problem 167
8.2.2. An Example Involving Multiple Explanations from an Intractable Theory 169
8.2.3. Discussion of Related Work Addressing the Multiple Explanations Problem 173
8.3. Scientific Discovery 173
8.3.1. Explanation-based Theory Revision: A Model for Scientific Theory Revision 175
8.3.2. An Example: Revision of the Phlogiston Theory of Combustion 176
8.3.3. Discussion of Related Work in Scientific Discovery 182
8.4. Summary 183
9 CONCLUSIONS 184
9.1. A Brief Review of Explanation-based Theory Revision 184
9.2. Limitations and Future Work 186
9.3. Related Work 188
9.4. Potential Applications 193
9.5. Significance of the Thesis 195

9.6. Summary 197
APPENDIX A DETAILS OF THE OSMOSIS EXAMPLE 198
A.1. Introduction 198
A.2. The Initial Theory 198
A.3. Explaining Observations Successfully 199
A.3.1. A Scenario Involving a Flow of a Fluid 200
A.3.2. A Scenario in which Flow Fails 200
A.3.3. Another Scenario in which Flow Fails 201
A.4. Episode 1: Learning a New Process 201
A.5. Episode 2: Correcting an Influence 210
A.6. Episode 3: Correcting Another Influence 221
A.7. Episode 4: Learning a New Quantity Condition 223
REFERENCES 229
VITA 237

CHAPTER 1

INTRODUCTION

1.1. Introduction

Knowledge-intensive systems are of increasing importance in Artificial Intelligence (AI). Such systems have been extensively researched in different areas of AI such as machine learning, qualitative reasoning, planning, and expert systems. Knowledge-intensive systems base their reasoning on a *domain theory* – an encoding of knowledge pertaining to the domain that is required by the system to insure its proper functioning. These systems apply the knowledge in the domain theory to make inferences about situations of interest from the domain. These inferences are used to guide different types of reasoning tasks such as explanation, prediction, diagnosis, classification, design and planning.

Systems that rely on a domain theory face two difficult problems. The performance of these systems is directly related to the amount of knowledge in the domain theory. In order to insure a satisfactory level of performance, the expert has the arduous and tedious chore of constructing a theory that has sufficient domain knowledge. He or she has to anticipate the rich variety of tasks and the wide range of situations for which the knowledge in the domain theory may be used. For many real world domains it is virtually impossible to handcode all the required knowledge. The domain expert is forced to take shortcuts, make approximations and omit knowledge. This results in an imperfect domain theory. Consequently, knowledge-intensive systems are either limited to a handful of domains that allow the construction of a good domain theory or are forced to operate with imperfect domain theories that can result in erroneous inferences.

The second difficult problem faced by knowledge-intensive systems is that they are fundamentally limited by the initial knowledge encoded in the domain theory. The reasoning of such systems is

confined to the deductive closure of the initial domain theory. The knowledge in the domain theory remains static and this leads to inflexible and inextensible systems that perform satisfactorily for a few examples but require large amounts of additional knowledge to process more examples. Knowledge–intensive systems that rely on static domain theories can quickly become obsolete. Consider a rapidly changing domain such as pathology. The theory of pathology is continually being revised as progress is made in the understanding of the diagnosis, prognosis, prevention, treatment and cure of diseases. Every day new vaccines and drugs are discovered; harmless stimulants are re–classified as carcinogens; beneficial drugs are shown to have pernicious side–effects; new diagnostic tests are discovered; more efficacious cures for diseases are discovered. A static domain theory for pathology will be rendered useless in no time.

The difficulty in building adequate domain theories and the inflexibility and inextensibility of static domain theories considerably restrict the applicability of knowledge–intensive systems. One solution to these problems is the construction of a *theory revision system* that automates the process of repairing and augmenting domain theories. This solution is desirable for two reasons: 1) It will free the expert of the laborious task of constructing a satisfactory domain theory. Instead, the expert can use "quick and dirty" methods to facilitate the construction of an operational, but not necessarily perfect, domain theory and can depend on the theory revision system to detect and correct problems with the theory as they are encountered. 2) The theory revision system can continually augment the theory to assimilate new knowledge that is discovered by investigating the failures of the original theory and through experimentation. Consequently, the system learns at the knowledge level [Dietterich86a], that is, the deductive closure of the knowledge in the theory changes due to the theory revision.

Figure 1.1 shows the architecture of the integrated system. The knowledge–intensive system uses the possibly imperfect, operational theory to reason about examples from the domain. The theory revision system monitors the performance of the knowledge–intensive system. When failures are encountered, the theory revision system augments or modifies the theory to eliminate the failures. As a result, the domain theory is continually and incrementally revised to conform with the observations made of the domain. This thesis describes a method for performing theory revision and a theory revision system implemented based on the method.

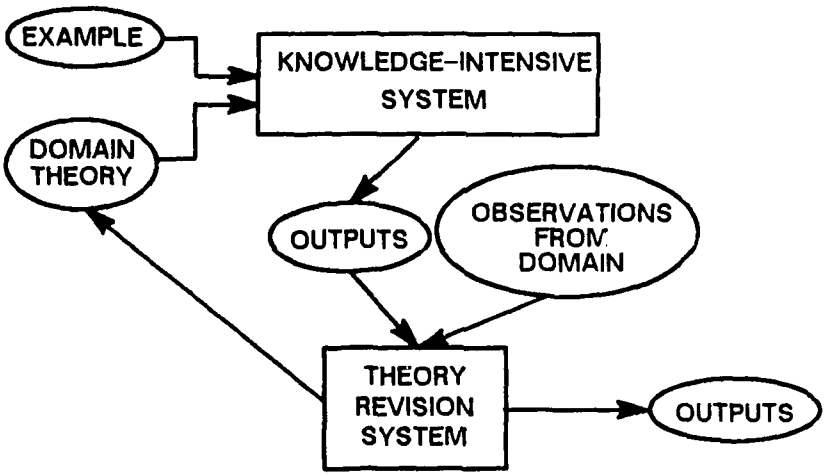


Figure 1.1 An architecture for the integrated system.

1.2. Knowledge-Intensive Systems

Figure 1.2 shows an inference engine that applies the knowledge in the domain theory to draw

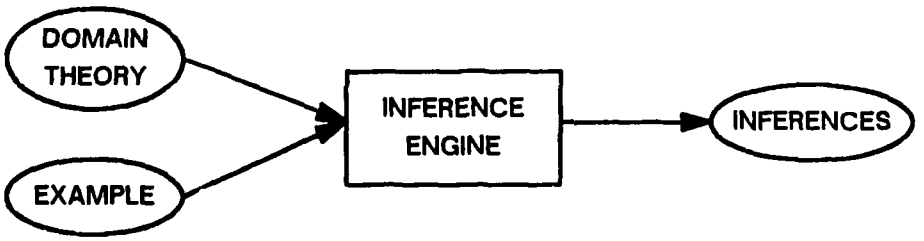


Figure 1.2 An inference engine of a knowledge-intensive system that applies the knowledge in the domain theory to draw inferences about the example.

inferences about examples from the domain. The inferences can be used by the knowledge-intensive system for various reasoning tasks such as predicting the behavior of the example, explaining observations, planning to achieve goals, etc. A failure occurs when expectations are violated: the predicted behavior may not be compatible with the observed behavior, the observations may not be explainable, the plan may not achieve the goal, etc. In

general, the failure can be traced to three sources: the example, the theory, or the inferencing procedure.

[a] Failure in the Example

The failure can be due to an incomplete or incorrect description of the example, the malfunctioning of objects in the example, false observations due to damaged sensors, etc. Research in *diagnosis* [Buchanan84, Davis84, de Kleer87, Genesereth84, Reiter87] addresses the problem of identifying a set of malfunctioning objects in the example that can explain the discrepancies in the observed behavior and the predicted behavior.

[b] Failure in the Theory

The failure can be due to inadequacies of the domain theory: relevant knowledge may be missing, some knowledge may be incorrect, the knowledge may be represented at too detailed a level, etc. *Theory formation* [Amarel86, Dietterich86b, Falkenhainer87a, Falkenhainer87b, Thagard85] addresses the problem of forming new theories of the domain. *Theory revision* [Ginsberg88, Rajamoney86a, Rajamoney88a, Shrager87] addresses the problem of augmenting or correcting an existing domain theory. *Theory approximation* [Bennett87, Chien87, Doyle86, Ellman88, Mostow87] addresses the problem of making approximations to simplify the process of drawing inferences.

[c] Failure in the Inferencing Mechanism

The failure can be due to problems with the inferencing mechanism: the inference procedure may be incomplete, the inference procedure may not be sound, etc. *Theorem proving* and *program verification* address some of the problems associated with inference procedures.

1.3. Overview of Explanation–based Theory Revision

This thesis addresses the problem of theory revision. It describes a method called *explanation–based theory revision* for augmenting and correcting domain theories. The method consists of five components:

1) Detecting problems with the theory by comparing the observations gathered from the domain with the predictions computed using the theory.

2) When problems occur, analyzing the problems and hypothesizing revisions to the domain theory.

3) Designing experiments to obtain additional information from the domain to eliminate some of competing hypotheses.

4) Rejecting proposed theories that cannot explain some of the previous observations that were successfully explained by the original theory.

5) Selecting a "best" theory from the remaining theories based on aesthetic criteria.

The thesis also describes COAST (for COrrecting and Augmenting Scientific Theories) – an implementation of explanation–based theory revision. COAST revises qualitative theories of physical domains when they fail to explain observations or make incorrect predictions.

Explanation–based theory revision focuses on problems with the domain theory. It assumes that the example is fully described and the observations supplied to the system are correct. It assumes that the inferencing mechanism is sound and complete – for example, it identifies all the feasible interactions and computes the interactions correctly.

1.4. Organization of the Thesis

Chapter 2 provides an overview of explanation–based theory revision and describes the implemented system, COAST. It also provides examples of the theory revision performed by COAST.

Chapters 3, 4, 5, 6, and 7 present a detailed description of each component method of explanation–based theory revision. Each chapter has a theoretical description followed by the implications for the implementation, COAST. Chapter 3 presents a taxonomy of the different types of problems with domain theories and describes general methods to detect these problems. Chapter 4 describes how hypotheses for revising the theory are generated. Chapters 5 and 6 discuss methods to test the proposed hypotheses. Chapter 5 describes a method for designing experiments to refute hypotheses. Chapter 6 describes a method for rejecting theories based on the previous observations of the system. Chapter 7 describes general criteria for selecting a theory from a number of competing theories.

Chapter 8 discusses additional applications of explanation-based theory revision and its component methods to problems in explanation-based learning and scientific discovery.

Chapter 9 discusses some of the limitations of explanation-based theory revision and the COAST system, identifies areas for future research, compares the approach to previous research in AI and outlines the major contributions of the thesis.

Appendix A describes in detail the examples of COAST's theory revision presented in section 2.3. Appendix A includes a description of the initial knowledge in the domain theory of the system and an annotated script of the output of the system.

CHAPTER 2

OVERVIEW OF EXPLANATION-BASED THEORY REVISION

2.1. Explanation-based Theory Revision

Explanation-based theory revision is a method for revising domain theories. The term *theory revision* or *revision* will be used to denote a change or an addition to the domain theory. The term *revised theory* will be used to denote a theory that is obtained by applying a set of revisions to the domain theory. Explanation-based theory revision consists of five component methods:

[a] Detection of problems

Problems with the domain theory are detected by comparing the predictions made by the domain theory with the observations made from the domain. Failures occur when the predictions are not compatible with the observations. There are three types of failures: 1) *broken explanations* – the observed behavior cannot be explained. 2) *contradictions* – the predictions contradict the observations. 3) *multiple explanations* – multiple, incompatible explanations are constructed for an observation. All these types of failures are symptoms of an inadequate domain theory.

[b] Hypothesis generation

When a failure occurs, revised theories, based on hypothesized revisions to the original domain theory, are proposed to eliminate the failure. The number of hypotheses generated is constrained by exploiting knowledge about the failure, the situation in which the failure occurred and the explanation construction process. Hypotheses are generated in two stages: 1) Abstract hypotheses are initially proposed to locate the parts of the theory that have to be revised. 2) These abstract hypotheses are tested and the successful hypotheses

are refined to concrete hypotheses that specify how the identified parts of the theory are to be revised.

[c] **Experimentation–based hypothesis refutation**

In general, there can be many different hypotheses that can eliminate the failure. Many of these will be incorrect. Experimentation–based hypothesis refutation is a method for testing competing hypotheses. The method involves designing experiments to obtain additional information from the domain. Hypotheses that are not consistent with the experimental observations are refuted.

[d] **Exemplar–based theory rejection**

Exemplar–based theory rejection is another method for testing hypotheses. This method insures that any newly proposed theory is consistent with previously observed phenomena. Selected examples, called *exemplars*, of previous observations that were successfully explained using the original theory are retained. The method rejects any proposed theory that cannot construct explanations for relevant exemplars.

[e] **Selection of a theory**

A "best" theory is selected from the remaining theories based on three criteria: the simplicity of the theory, the simplicity of the explanations constructed by the theory and the predictive power of the theory.

A domain theory is amenable to revision by explanation–based theory revision if it satisfies the following requirements:

[a] The domain must have observable features and the values of the features must be such that contradictions and compatibility among the values are easy to determine. These requirements are necessary for the effective detection of problems with the domain theory and the design of useful experiments.

[b] The domain representation must be such that a finite set of types of revisions for the different components of the theory can be identified. This insures that the hypothesis generation process can construct the correct revised theory.

[c] The domain must be manipulable. This requirement is necessary for the construction of experiments that alter the examples from the domain.

[d] The domain must be decomposable into components with specific functionality. This requirement is necessary for exemplar-based theory rejection which retains and retrieves exemplars of components of the theory according to their functionality.

2.2. COAST

COAST is an implementation of explanation-based revision for revising qualitative theories of physical domains. COAST addresses the tasks of constructing explanations for observations from the physical world or making predictions about the behavior of the physical world. Figure 2.1

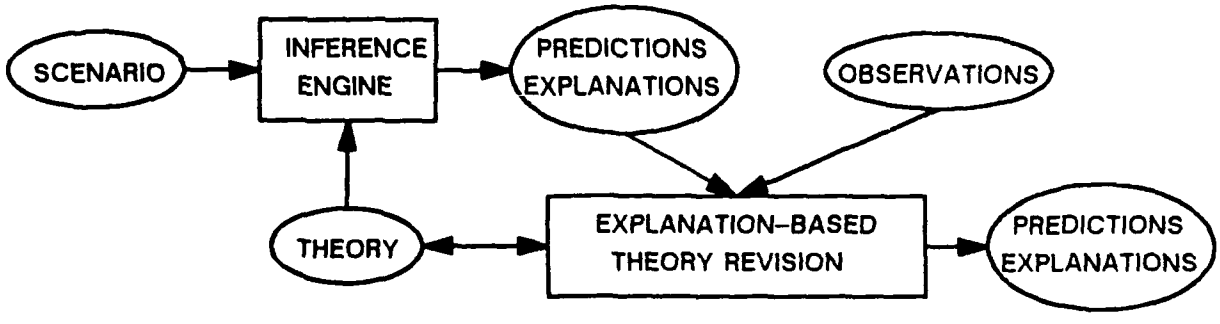


Figure 2.1 The architecture of the COAST system.

illustrates the architecture of the system. COAST employs an inference engine to compute the behavior of a situation of interest from the domain. Explanation-based theory revision compares the behavior predicted by the inference engine with the observed behavior. If failures are encountered the initial theory is inadequate and must be revised. COAST uses explanation-based theory revision to produce a revised theory that eliminates the failures. The revised theory is used to determine the correct behavior for the given situation and other situations subsequently encountered.

2.2.1 Knowledge Representation in COAST

There are three types of knowledge descriptions used by the system - a *domain theory* that describes the knowledge in the domain, a *scenario* that describes a situation of interest from the domain and *observations* that specify known changes in the scenario.

[a] Domain Theory

Domain theories are represented using Forbus' Qualitative Process (QP) theory [Forbus84a, Forbus84b]. QP theory attempts to capture the common sense physical reasoning that people perform when reasoning about changes in the physical world. There are many different kinds of changes in the physical world – objects move, heat up, cool down, boil, dissolve, stretch and break. QP theory characterizes these and similar changes as due to *processes*.

QP theory models continuous properties of objects such as the temperature or pressure of a liquid by *quantities*. A quantity consists of an *amount* and a *derivative* – both of which are *numbers*. QP theory does not deal with numerical values for these quantities. Instead, a number is described in terms of the inequalities with other numbers. QP theory computes and reasons with qualitative values such as an increase or a decrease in the value of a quantity.

A process consists of five pieces of information: *individuals* – objects and processes that participate in the process; *preconditions* and *quantity conditions* – conditions that must be satisfied for the process to be active; and, *relations* and *influences* – statements that hold if the process is active. A *process instance* is formed whenever there are objects and processes meeting the specifications in the individuals field; the process instance is active if all the preconditions and quantity conditions are satisfied; and, all the statements in the relations and influences fields hold when the process instance is active. Figure 2.2 shows the definition of a process for the flow of fluids.

Individuals:

> The individuals field specifies the set of objects and processes that participate in the process and the requirements that must be satisfied by each participant[1]. In the example of the flow process, there are three individuals – a source and a destination, both of which must be contained fluids, and a path which must be a fluid path connecting the source and the destination. When a collection of objects satisfying these specifications are available in the scenario, a process instance is formed corresponding to the possibility of a flow from the source to the destination through the path.

[1] These requirements specify the type of each individual. Other constraints can also be specified but, in theory, these constraints are a part of the preconditions of the process. The purpose of including additional constraints in the individuals' specifications is to limit the number of candidate collection of objects that must be considered

```
Fluid Flow (?source ?destination ?path)
  Individuals
    ?source              (contained-fluid ?source)
    ?destination         (contained-fluid ?destination)
    ?path                (fluid-path ?path)
                         (path-connection ?source ?destination ?path)
  Preconditions
    (fluid-flow-aligned? ?path)
  Quantity Conditions
    (greater-than (A (pressure ?source)) (A (pressure ?destination)))
  Relations
    (Q+ (fluid-flow-rate ?self) (pressure ?source))
    (Q- (fluid-flow-rate ?self) (pressure ?destination))
    (Q+ (fluid-flow-rate ?self) (cross-sectional-area ?path))
    (Q- (fluid-flow-rate ?self) (length ?path))
  Influences
    I-[(amount-of ?source), (A (fluid-flow-rate ?self))]
    I+[(amount-of ?destination), (A (fluid-flow-rate ?self))]
```

Figure 2.2 A description of the process for the flow of fluids ("A" stands for the amount of the quantity).

Preconditions and Quantity Conditions:

These are conditions that must be satisfied before the process instance can become active. Quantity conditions specify the inequalities between quantities that must hold or the processes that must be active. In the flow process, the pressure at the source must be greater than the pressure at the destination. Preconditions specify all the other conditions that must be satisfied. The distinction between preconditions and quantity conditions is that QP theory can determine and reason about the status of quantity conditions whereas it cannot for preconditions. The flow process has a precondition which specifies that the fluid path must be aligned, that is, all the valves in the path must be open.

Relations:

The relations field specifies the statements other than the influences that hold when the process instance is active. It includes the functional dependency between quantities. These are of the form:

$(Q+ Q_1 Q_2)$ or $(Q- Q_1 Q_2)$

depending on whether Q_1 is strictly increasing or decreasing in its dependence on Q_2. In the flow process, the rate at which the flow occurs is qualitatively proportional to the pressures at the source and the destination and to the length and cross-sectional area of the path. The

relations field also specifies the inequality relations between quantities that are true as a consequence of the process being active[2].

Influences:

Influences specify the direct effects of a process. An influence is of the form:

I+[Q, n] or I−[Q, n]

depending on whether the number n influences the quantity Q positively or negatively. If a quantity is directly influenced, the change in the quantity is computed by combining all the positive and negative influences on the quantity. In the flow example, the direct effects of an active flow process instance are a positive influence on the amount of the liquid at the source and a negative influence on the amount of the liquid at the destination. If the flow process is the only active process then the amount of the liquid at the source decreases and the amount of the liquid at the destination increases. A quantity Q changes only if it is directly influenced or if it is indirectly influenced through a qualitative proportionality.

In QP theory, objects, such as liquid in a container, whose existence depends on dynamical constraints, are described as individual views. Individual views are represented in a fashion similar to the process definitions. These views are specified by individuals, preconditions, quantity conditions and relations. However, individual views do not have the influences component – only processes can influence quantities. Figure 2.3 describes an individual view for a solution that is composed of a solute and a solvent. The precondition specifies that the solute must be soluble in the solvent. The relations state that the concentration of the solution is proportional to the amount of the solute and inversely proportional to the amount of the solvent.

A domain represented in QP theory consists of a description of all the processes and individual views in the domain. A domain theory for physical domains involving liquids includes process definitions such as boiling, cooling, heating, evaporation, condensation, absorption, flow and dissolving and includes definitions of individual views such as solutions, liquids, solids and gases.

[2] COAST currently allows only qualitative proportionalities and inequality relations between quantities to be specified in the relations field. QP theory allows other types of information to be specified in the relations field such as correspondences which map information about inequalities across qualitative proportionalities.

```
Solution (?solution)
   Individuals
      ?solution    (contained-liquid ?solution)
   Preconditions
      (soluble? (solute-of ?solution) (solvent-of ?solution))
   Quantity Conditions
      (greater-than (A (amount-of (solute-of ?solution))) 0)
   Relations
      (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution)))
      (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))
      (Q- (concentration ?solution) (amount-of (solvent-of ?solution)))
```

Figure 2.3 A description of a solution formed by dissolving a solute in a solvent.

Note that there is no canonical representation for a domain in QP theory. QP theory, and other qualitative reasoning representations, permit multiple representations for a domain. This has implications for a theory revision system. In particular, the revised theories will not converge to a canonical description of the domain because one does not exist. Instead, a theory revision system converges if it produces revised theories whose predictions conform to the observations made from the domain.

[b] Scenario

A *scenario* is a description of a situation of interest from the physical domain. A scenario is composed of two parts: a layout and a behavior. The *layout* of a scenario specifies the objects in the scenario; the static physical distribution of the objects – the particular manner in which the objects are organized in the situation; and, the initial relationships between quantities of the objects. For example, the layout specifies whether a path is aligned for a flow of fluids, a path connects liquid in two containers, a piece of salt is soluble in water, a container is open, the temperature of a liquid is initially less than its boiling point, and a piece of sponge absorbs water.

Figure 2.5 shows a partial description of the layout of the scenario shown in figure 2.4. In this scenario there are three individuals – water in container1, water in container2 and a pipe connecting the two volumes of water. The pipe is aligned for fluid flow; the two containers are open; and, the pressure of water in container1 is greater than the pressure of water in container2.

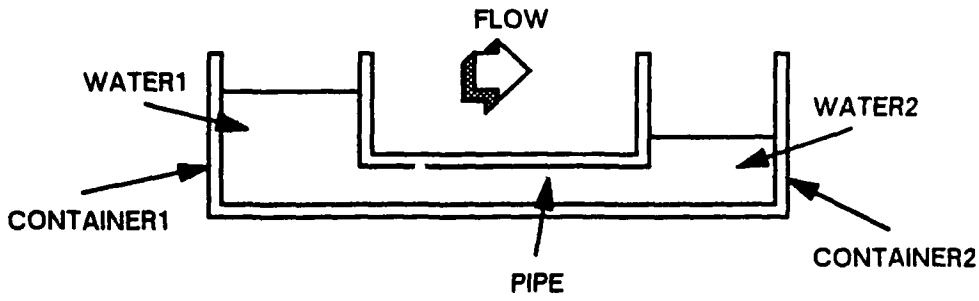


Figure 2.4 A scenario in which water in two containers is connected by a pipe.

```
Two-container-scenario:
   Individuals:
      water1 water2 pipe
   Facts:
      (fluid-path pipe)
      (fluid-connection water1 water2 pipe)
      (fluid-flow-aligned pipe)
      (open (container water1))
      (open (container water2))
      (greater-than (A (pressure water1)) (A (pressure water2)))
```

Figure 2.5 A partial description of the layout of the scenario shown in figure 2.4.

The *behavior* of a scenario specifies the changes to the quantities and the dynamic relationships between the quantities in the scenario. COAST computes the behavior of the scenario from the layout and the domain theory using an inference engine based on the algorithms described by Forbus [Forbus84b]. The inference engine used by COAST is a substantially simplified version of Forbus' inference engine. In particular, it currently does not perform *limit analysis* [. forbus phd .] – that is, determining the changes to the processes and individuals due to the changes in the quantities. Consequently, unlike Forbus' inference engine, it can not compute the *envisionment* of a scenario – that is, the different qualitative states into which the initial state of the scenario can evolve over time[3]. In the subsequent discussion it will be assumed that the behavior of a scenario refers to the behavior within the initial qualitative state.

The inference engine predicts the changes to each quantity in the scenario – whether the quantity increases, decreases or remains constant. It also constructs explanations for each of these

[3] A qualitative state refers to a unique collection of processes, individuals, and values for the changes to the quantities.

predictions. Figure 2.6 shows some of the predictions and explanations for the scenario in figure 2.4.

```
Predicted Changes:
   (Decrease (amount-of water1))
   (Increase (amount-of water2))

Explanations:
   (Decrease (amount-of water1))
      I-[(amount-of water1), (A (Flow-rate (Flow water1 water2 pipe)))]
         Active (Flow water1 water2 pipe)
            (Fluid-flow-aligned? pipe)
            (greater-than (A (pressure water1)) (A (pressure water2)))
   (Increase (amount-of water2))
      I+[(amount-of water2), (A (Flow-rate (Flow water1 water2 pipe)))]
         Active (Flow water1 water2 pipe)
            (Fluid-flow-aligned? pipe)
            (greater-than (A (pressure water1)) (A (pressure water2)))
```

Figure 2.6 Some of the predictions and explanations for the scenario of figure 2.4.

[c] Observation

Changes known to occur in the scenario are specified as observations. Observations are of the form:

```
(?change ?quantity)
```

where the change may be an increase, decrease or constant and the quantity is any continuous property of an object specified in the theory. The observations represent changes within a single qualitative state. Examples are:

```
(increase (amount-of water))
(decrease (temperature alcohol))
(constant (pressure vapor)).
```

2.2.2. Satisfying the Requirements for Explanation-based Theory Revision

Each of the four requirements for applying explanation-based theory revision are satisfied by theories of physical domains represented in QP theory:

[a] QP theory represents the changes in quantities by three values: increase, decrease and constant. Compatibility and contradictions among these values are easy to determine.

Changes in many of the quantities (the continuous properties of objects) are readily observable or measurable in the physical world.

[b] A finite set of types of revisions for the processes, individuals and their components can be identified. These are: adding a new component, deleting an existing component, narrowing the scope of a component, widening the scope of a component and inverting a component.

[c] Physical domains are manipulable. Therefore, experiments can be designed to alter the conditions of a situation.

[d] Processes and individual views can be decomposed into individuals, preconditions, quantity conditions, relations and influences. Each of these components serves a well-defined function. For example, the influences of a process represent the direct effects of the process and the quantity conditions of a process represent the inequalities among the quantities and the status of processes that must be satisfied before the process can become active.

2.2.3. Examples Implemented in COAST

COAST has been demonstrated on ten fully implemented examples of theory revision. Four of these examples involve the learning and the revision of process descriptions for osmosis. These four examples are described in the next section. Further details and a trace of COAST's output for these examples are provided in appendix A. Four other examples deal with the revision of qualitative theories of the evaporation of a liquid, the dissolving of a substance in a liquid, and the flow of a fluid. These four examples are used to illustrate each step of explanation-based theory revision and are discussed in chapters 3, 4, 5, 6 and 7. The ninth example involves the revision of a qualitative representation of the phlogiston theory of combustion and is described in chapter 8. The tenth example is from chemistry and involves the chemical decomposition of water into oxygen and hydrogen. This example is described in chapter 8.

2.3. Examples of COAST's Theory Revision

This section presents examples of the theory revision performed by COAST (further details of these examples including a detailed, annotated trace of the output of COAST are presented in appendix A). The examples are from the domain of liquids. The initial domain theory (figure 2.7) consists of processes describing the flow of liquids, evaporation of liquids, condensation of vapor,

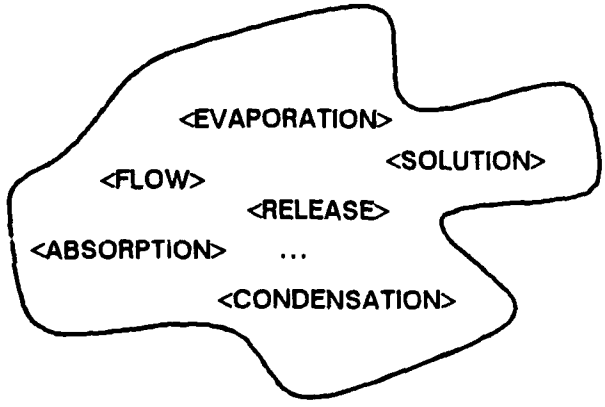


Figure 2.7 A partial description of the initial theory of liquids used by the system.

absorption of liquids by solids and release of the absorbed liquid by solids. It also has individual views describing substances such as solutions. The system uses this domain theory to predict the behavior of and explain the observations made in the scenarios drawn from the liquids domain such as water in two containers connected by a pipe, alcohol placed in an open container, and a sponge placed in contact with some water.

[a] Learning a new process

The system is given the scenario shown in figure 2.8 and asked to explain an observed decrease in the amount of the solution in the first container. In the scenario, two solutions of different concentrations are placed in containers that are separated by a partition. Unknown to the system,

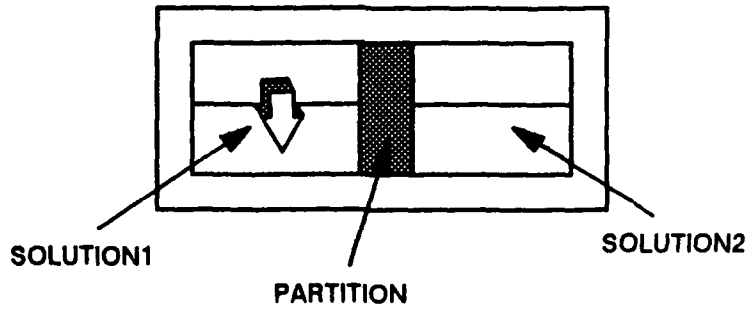


Figure 2.8 A scenario in which two solutions of different concentrations are placed in two separate containers connected by a partition. The amount of solution1 is observed to be decreasing.

the partition is *semi-permeable* and a process called *osmosis* takes place. Osmosis is the flow of solvent through a semi-permeable path from the solution of lower concentration to the solution of higher concentration. The system has no prior knowledge of osmosis or semi-permeability.

```
(explain-observation '(decrease (amount-of Solution1)))

Cannot explain observation. invoking theory revision ...

Hypothesizing revisions ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Refining hypotheses ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Selecting theory ...
Finished theory revision.

New process is:

Process8974 (?var8975 ?var8976 ?var8977)
   Individuals:
      ?var8975   (contained-fluid ?var8975) (contained-liquid ?var8975)
      ?var8976   (contained-fluid ?var8976) (contained-liquid ?var8976)
      ?var8977   (path ?var8977)
   Preconditions:
      (precondition8978 ?var8975 ?var8976 ?var8977)
   Quantity Conditions:
   Relations:
      (Q+ (process8974-rate ?self) (cross-sectional-area ?var8977))
   Influences:
      I+[(amount-of ?var8976), (A (process8974-rate ?self))]
      I-[(amount-of ?var8975), (A (process8974-rate ?self))]

Explanation for (decrease (amount-of Solution1))

(decrease (amount-of Solution1))
   I-[(amount-of Solution1), (A (process8974-rate Solution1 Solution2 Partition))]
      (Active (process8974 Solution1 Solution2 Partition))
         (precondition8978 Solution1 Solution2 Partition)
```

The behavior computed by the system based on the initial theory predicts that no changes will occur in the given scenario. This is because all the feasible processes are inactive due to failed conditions: flow cannot occur because the paths through the partition and the wall of the container are not aligned for fluid flow; evaporation and condensation do not occur because both containers are closed; release and absorption do not occur because both the solids involved – the wall of the containers and the partition – are not absorbent. Therefore, the initial theory cannot explain the decrease in the amount of the solution in the first container. The system then invokes theory revision to revise the domain theory.

COAST generates hypotheses to eliminate the failure (such as abstract hypotheses like a flow through the wall occurs even though the wall is not aligned or evaporation occurs despite the closed container and concrete hypotheses like the fluid-flow-aligned precondition is not required for flow), designs experiments to test these hypotheses, rejects revised theories based on exemplars and ultimately arrives at a revised theory (figure 2.9) that is formed by adding a new process to the

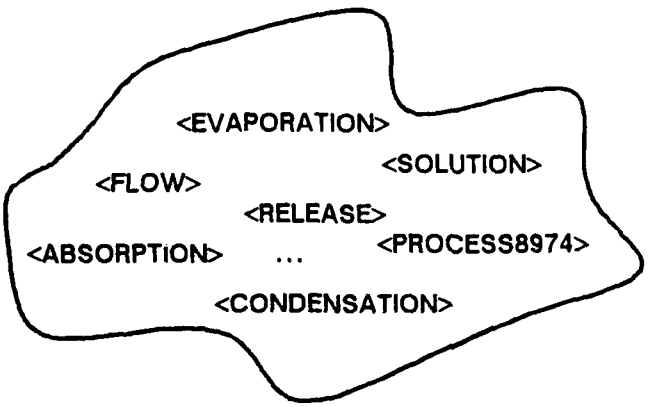


Figure 2.9 A revised theory formed by adding a new process to the theory of figure 2.7.

original theory. The new process has three participants - two contained solutions and a fluid path connecting the two solutions. A new precondition is postulated for the process. The rate of the new process is qualitatively proportional to the cross-sectional area of the path. The direct influences are a decrease in the amount of the solution at the source and an increase in the amount of the solution at the destination. This new theory can explain the observed decrease in the amount of the solution in the first container.

There are two important points to be noted about the revised theory. First, the new process incorporates information that is obtained during the experimentation phase of the theory revision. For example, the increase in the amount of Solution2 or the qualitative proportionality between the process rate and the cross-sectional area of the path were not initially specified. The explanation for the experimentally discovered increase in the amount of solution2 is given below.

```
(explain-observation '(increase (amount-of Solution2)))

Explanation for (increase (amount-of Solution2))

(increase (amount-of Solution2))
   I+[(amount-of Solution2), (A (process8974-rate Solution1 Solution2 Partition))]
      (Active (process8974 Solution1 Solution2 Partition))
         (precondition8978 Solution1 Solution2 Partition)
```

Second, though the revised theory has successfully explained the observed changes, it is not a correct or complete representation of the osmosis process - osmosis involves a flow of solvent whereas the new process describes a flow of solution; osmosis is active only if there is a difference in the concentrations of the two solutions; the rate of osmosis depends on the concentrations of the two solutions etc. Instead, COAST learns an initial, imperfect version of osmosis that suffices to explain the observed behavior of the scenario. This imperfect theory is revised as more failures are encountered.

[b] Correcting an influence

```
(explain-observation '(increase (concentration Solution1)))

Cannot explain observation. Invoking theory revision ...

Hypothesizing revisions ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Refining hypotheses ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Selecting theory ...
Finished theory revision.

Revised process is:

Process8974 (?var8975 ?var8976 ?var8977)
   Individuals:
      ?var8975   (contained-fluid ?var8975) (contained-liquid ?var8975)
      ?var8976   (contained-fluid ?var8976) (contained-liquid ?var8976)
      ?var8977   (path ?var8977)
   Preconditions:
      (precondition8978 ?var8975 ?var8976 ?var8977)
   Quantity Conditions:
   Relations:
      (Q+ (process8974-rate ?self) (cross-sectional-area ?var8977))
   Influences:
      I+[(amount-of ?var8976), (A (process8974-rate ?self))]
      I-[(amount-of (solvent-of ?var8975)), (A (process8974-rate ?self))]
```

Explanation for (increase (concentration Solution1))

```
(increase (concentration Solution1))
   (decrease (amount-of (solvent-of Solution1)))
      I-[(amount-of (solvent-of Solution1)),
                A (process8974-rate Solution1 Solution2 Partition))]
         (Active (process8974 Solution1 Solution2 Partition))
            (precondition8978 Solution1 Solution2 Partition)
   (Q- (concentration Solution1) (amount-of (solvent-of Solution1)))
      (Active (solution Solution1))
         (greater-than (A (amount-of (solute-of Solution1))) 0)
         (soluble? (solute-of Solution1) (solvent-of Solution1))
```

The system is next asked to explain an observed increase in the concentration of Solution1 in the same scenario. The revised theory fails to explain this observation. According to this theory, the only active process is the newly acquired process – process8974. However, this process does not affect the concentration of Solution1 because, according to the process, the solution as a whole flows through the path. Theory revision is invoked to deal with this failure. After eliminating a number of competing hypotheses, COAST accepts a revised theory in which the amount of the solvent of the solution is negatively influenced by process8974. This revised theory can explain the observed increase in the concentration of the solution.

There are two important points to note about the revision. First, unlike the first revision which dealt with an incomplete theory, this revision deals with an incorrect theory. Second, exemplar-based theory refutation, a part of the theory revision, insures that the revised theory can explain the previously encountered observations. For example, the revised theory can still explain the previously observed decrease in the amount of the solution (see below). However, the explanation is significantly different due to the revisions to the old theory.

Explanation for (decrease (amount-of Solution1))

```
(decrease (amount-of Solution1))
   (decrease (amount-of (solvent-of Solution1)))
      I-[(amount-of (solvent-of Solution1)),
                (A (process8974-rate Solution1 Solution2 Partition))]
         (Active (process8974 Solution1 Solution2 Partition))
            (precondition8978 Solution1 Solution2 Partition)
   (Q+ (amount-of Solution1) (amount-of (solvent-of Solution1)))
      (Active (solution Solution1))
         (greater-than (A (amount-of (solute-of Solution1))) 0)
         (soluble? (solute-of Solution1) (solvent-of Solution1))
```

[c] Correcting another influence

```
(explain-observation '(decrease (concentration Solution2)))

Cannot explain observation. Invoking theory revision ...

Hypothesizing revisions ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Refining hypotheses ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Selecting theory ...
Finished theory revision.

Revised process is:

Process8974 (?var8975 ?var8976 ?var8977)
   Individuals:
      ?var8975   (contained-fluid ?var8975) (contained-liquid ?var8975)
      ?var8976   (contained-fluid ?var8976) (contained-liquid ?var8976)
      ?var8977   (path ?var8977)
   Preconditions:
      (precondition8978 ?var8975 ?var8976 ?var8977)
   Quantity Conditions:
   Relations:
      (Q+ (process8974-rate ?self) (cross-sectional-area ?var8977))
   Influences:
      I+[(amount-of (solvent-of ?var8976)), (A (process8974-rate ?self))]
      I-[(amount-of (solvent-of ?var8975)), (A (process8974-rate ?self))]

Explanation for (decrease (concentration Solution2))

(decrease (concentration Solution2))
   (increase (amount-of (solvent-of Solution2)))
      I+[(amount-of (solvent-of Solution2)),
                A (process8974-rate Solution1 Solution2 Partition))]
         (Active (process8974 Solution1 Solution2 Partition))
            (precondition8978 Solution1 Solution2 Partition)
   (Q- (concentration Solution2) (amount-of (solvent-of Solution2)))
      (Active (solution Solution2))
         (greater-than (A (amount-of (solute-of Solution2))) 0)
         (soluble? (solute-of Solution2) (solvent-of Solution2))
```

The system is next asked to explain an observed decrease in the concentration of Solution2 in the same scenario. The revised theory fails to explain this observation. According to this theory, the only active process is the newly acquired process - process8974. However, according to the description of this process in the theory, the process does not affect the concentration of Solution2 because the amount of the entire solution increases. Theory revision is invoked to deal with this failure. After eliminating a number of competing hypotheses, COAST accepts a revised theory in

which the amount of the solvent of the destination solution is positively influenced by process8974. This revised theory can explain the observed decrease in the concentration of solution2.

[d] Learning a new quantity condition

```
(observation '(constant (amount-of Solution1)))

Prediction not compatible with observation. Invoking theory revision ...

Hypothesizing revisions ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Refining hypotheses ...
Experimentation-based hypothesis refutation ...
Exemplar-based theory rejection ...
Selecting theory ...
Finished theory revision.

Revised process is:

Process8974 (?var8975 ?var8976 ?var8977)
   Individuals:
      ?var8975   (contained-fluid ?var8975) (contained-liquid ?var8975)
      ?var8976   (contained-fluid ?var8976) (contained-liquid ?var8976)
      ?var8977   (path ?var8977)
   Preconditions:
      (precondition8978 ?var8975 ?var8976 ?var8977)
   Quantity Conditions:
      (greater-than (A (concentration ?var8976)) (A (concentration ?var8975)))
   Relations:
      (Q+ (process8974-rate ?self) (cross-sectional-area ?var8977))
   Influences:
      I+[(amount-of (solvent-of ?var8976)), (A (process8974-rate ?self))]
      I-[(amount-of (solvent-of ?var8975)), (A (process8974-rate ?self))]

Prediction: (constant (amount-of Solution1))
```

The system is next asked to predict the behavior of the scenario shown in figure 2.10. This scenario is exactly the same as that of figure 2.8 except that the concentrations of the two solutions are now equal. Based on the revised theory, the system predicts that the process8974 is active and consequently the amount of Solution1 is decreasing. However, the amount is observed to remain constant. This failure invokes theory revision again. After eliminating many hypotheses COAST finally arrives at a revised theory that incorporates a new quantity condition into process8974. The new quantity condition requires the concentration of the destination solution to be greater than the concentration of the source solution. This revised theory correctly predicts that the amount of the solution in the second scenario does not change because none of the processes influencing the

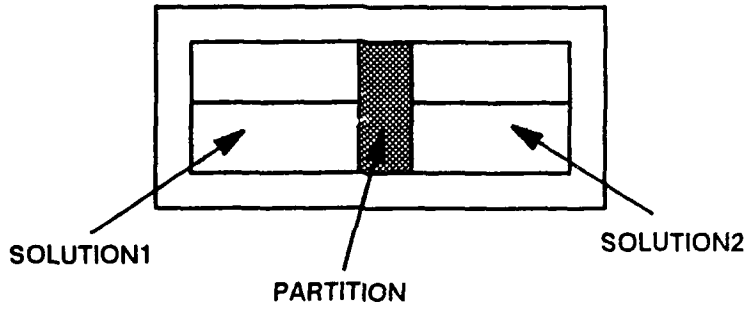


Figure 2.10 A scenario similar to the scenario of figure 2.8 except that the two solutions have equal concentrations.

amount are active. Process8974 is no longer predicted to be active since the new quantity condition is not satisfied.

As a result of the series of failures and theory revision, the system has learned a new process that corresponds to osmosis. The process description is not yet accurate because it does not include qualitative proportionalities involving the rate of the process with respect to the difference in concentrations, the length of the path etc. Further failures have to be encountered before the system will have learned a complete qualitative description of osmosis.

2.4. Summary

This chapter presented an overview of explanation-based theory revision and the COAST system. A brief description of each component method of explanation-based theory revision – the detection of problems, hypothesis generation, experimentation-based hypothesis refutation, exemplar-based theory rejection, and the selection of theories – was provided. The architecture and knowledge representation of the COAST system were described. Finally, examples of the theory revision performed by COAST on incomplete and incorrect domain theories were described.

CHAPTER 3

THE CLASSIFICATION AND DETECTION OF IMPERFECT THEORY PROBLEMS

3.1. Introduction

This chapter begins the theoretical analysis of revising domain theories. It discusses the different types of problems with domain theories and methods for detecting these problems. The second section discusses taxonomies for classifying problems with the domain theories. The third section illustrates some of these problems with examples from QP theory. The fourth section describes different methods for detecting these problems. The fifth section discusses a general mechanism for detecting some of these problems and uses examples from QP theory to illustrate the mechanism. The last section discusses research related to the classification and detection of imperfect theory problems.

3.2. The Classification of Imperfect Theory Problems

Mitchell et al. [Mitchell86] have briefly classified problems with imperfect domain theories in the context of explanation–based generalization into three categories:

(1) the *incomplete theory problem*: the domain theory is not complete enough to construct explanations.

(2) the *intractable theory problem*: the domain theory is complete, but it is computationally very expensive to construct explanations using the domain theory.

(3) the *inconsistent theory problem*: inconsistent statements can be derived from the theory.

However, the underlying issues are too murky and subtle for the above categories to be cleanly separable. For example, inconsistencies and incompleteness in domain theories can be due to

abstractions and approximations which make an intractable theory tractable [Doyle86]. Inconsistent theory problems can be due to an incomplete theory if information necessary to nullify one of the inconsistent statements is missing. Inconsistent statements can also result from the incomplete theory problem if the closed world assumption is made and the possibility of new information influencing the computations is not considered [Rajamoney86a]. Apart from the above problems of interacting categories, the classification of Mitchell et al. also ignores certain kinds of incompleteness and inconsistencies.

A complete taxonomy of imperfect theory problems [Rajamoney87] includes two types of incompleteness, two types of intractability, and incorrectness.

Incompleteness – Type I:

The first type of incompleteness is the one discussed above in which a desired conclusion cannot be obtained because the deductions leading to the conclusion cannot be completed. Knowledge required to complete the deductions is missing from the domain theory.

Incompleteness – Type II:

The second type of incompleteness is due to the lack of sufficient detail in the relevant knowledge. Unlike the first case, deductions can be constructed leading to a conclusion. However, due to the lack of detail, assumptions have to be made and this leads to multiple, incompatible explanations for a conclusion.

Intractability – Type I:

The first type of intractability is due to large search spaces – the explanation cannot be constructed within the allotted resources, even though the explanation exists and its size is comparable to previous explanations, because there are too many choices at each decision point in the explanation construction process. The resources are not sufficient to exhaustively search every choice. The problem with the domain theory is that it has very little or ill-specified control knowledge.

Intractability – Type II:

The second type of intractability is a consequence of a very detailed representation of the domain theory. The explanation cannot be finished because it is too large to be constructed

within the given resources. This problem is independent of the search problem described above – even if there is no search, the explanation can still be too large to be constructed.

Incorrectness:

Knowledge in the domain theory is incorrect and the incorrect knowledge has to be identified and revised. The incorrect knowledge can result in an internally inconsistent domain theory as in Mitchell et al.'s classification. In this case, inconsistent statements can be derived from the domain theory. Or the incorrect knowledge can result in a domain theory that yields conclusions that are not consistent with the observations made from the domain. In the latter case, the domain theory can still be internally consistent.

Table 3.1 summarizes these five types of problems with the domain theories.

TYPE	FAILURE	PROBLEM	REVISION
INCOMPLETENESS TYPE – I	Cannot deduce a given conclusion.	Missing knowledge.	Add knowledge to complete the deduction.
INCOMPLETENESS TYPE – II	Multiple, incompatible explanations for a given conclusion.	Insufficient detail.	Add knowledge to determine the correct explanation.
INTRACTABLE TYPE – I	Insufficient resources to completely deduce a given conclusion.	Large search space.	Add knowledge to focus the search.
INTRACTABLE TYPE – II	Insufficient resources to completely deduce a given conclusion.	Inappropriate level of detail in the domain theory.	Change the level of representation, make approximations, assumptions.
INCORRECTNESS	Deduces a false conclusion or inconsistent statements.	Incorrect knowledge.	Identify incorrect knowledge and correct it.

Table 3.1 The five different types of problems with the domain theories.

3.3. Examples of Incomplete and Incorrect Domain Theories in QP Theory

This section presents examples that illustrate the different types of incomplete and incorrect domain theories described in the previous section. The examples use domain theories that are represented

in QP theory. Intractable domain theories are not further discussed in this chapter – chapter 8 contains a section describing problems related to intractable domain theories.

Incompleteness – Type I:

In this type of incompleteness, the domain theory does not have the necessary components required to explain an observation. New components such as new processes, individuals, preconditions, quantity conditions, relations and influences are added to the theory to revise this type of incompleteness.

For example, consider the small part of the liquids domain shown in figure 3.1. The theory describes a process definition for the evaporation of liquids. According to this definition, the amount of the liquid in an open container decreases and the amount of its vapor increases due to evaporation.

```
Evaporation (?liquid ?vapor)
   Individuals
      ?liquid ?vapor
   Preconditions
      (open? (container ?liquid))
   Quantity Conditions
   Relations
      (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))
   Influences
      I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
      I+[(amount-of ?vapor), (A (evaporation-rate ?self))]
```

Figure 3.1 A part of the domain theory for liquids that describes the evaporation of liquids.

Figure 3.2 shows a scenario in which alcohol is placed in an open container. In addition to the

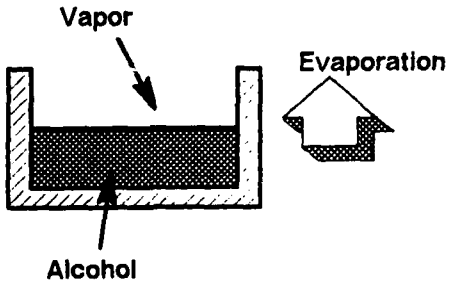


Figure 3.2 A scenario in which alcohol is placed in an open container. Alcohol vapor is in contact with the alcohol.

changes in the amounts of the alcohol and its vapor, the temperature of the alcohol is also observed

to decrease. The domain theory described in figure 3.1 cannot explain this observed decrease because it is incomplete. It does not have the required knowledge to associate evaporation of alcohol with the observed drop in the temperature of alcohol. The theory can be revised by adding a new influence to the evaporation process definition such as:

I-[(temperature ?liquid), (A (evaporation-rate ?self))].

This revised theory can explain the observed decrease in the temperature of alcohol as a direct effect of the evaporation of alcohol which is active in the scenario.

Incompleteness – Type II:

In this type of incompleteness, the domain theory does not have knowledge to determine which of a set of choices is the correct one and is therefore forced to assume each one. Explanations based on each assumption can be constructed. Adding new facts to the theory can eliminate the choice or reduce the set of choices.

For example, consider the scenario shown in figure 3.3. In this scenario, a mixture of water and

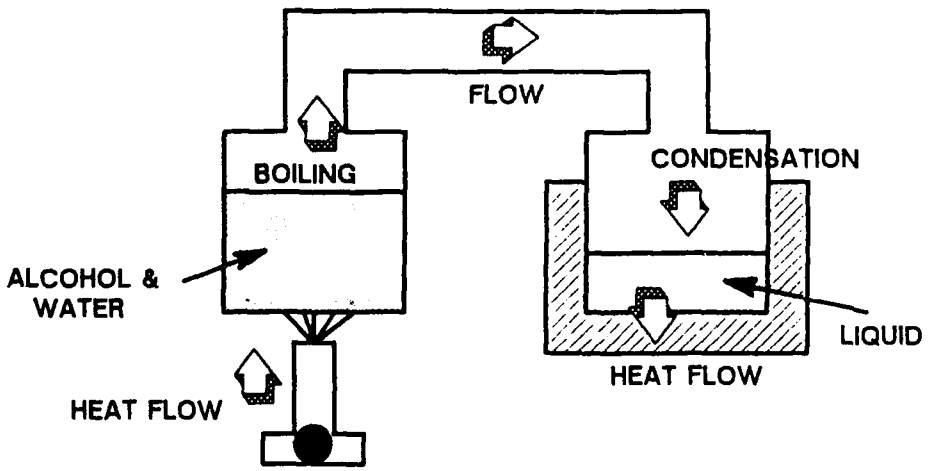


Figure 3.3 A scenario in which a mixture of alcohol and water is heated. The container is connected to another container which is maintained at a very low temperature.

alcohol is heated in a container. One of the two liquids boils and the vapor flows through the pipe into the second container where it condenses on coming into contact with the cold liquid. A single explanation for the observed increase in the amount of liquid in the second container cannot be

constructed because the domain theory is incomplete. It does not include relevant knowledge about the boiling points of water and alcohol – whether the boiling point of alcohol is greater than, less than or equal to that of water. Therefore, there are three explanations for the amount of liquid increasing in the second container – 1) The boiling point of alcohol is less than that of water and it boils first. Under this assumption, the liquid in the second container is alcohol. 2) The boiling point of alcohol is greater than that of water and water boils first. Under this assumption, the liquid in the second container is water. 3) The boiling point of alcohol is equal to that of water and both boil together. Under this assumption, the liquid in the second container is a mixture of alcohol and water. The incomplete theory can be revised by adding new knowledge about the relationship between the boiling points of water and alcohol such as:

Boiling-point(alcohol) < Boiling-point(water).

This will result in a single explanation for the observed increase in the amount of liquid in second container.

Incorrectness:

Some components of the domain theory are incorrect and this leads to false predictions. The incorrect components have to be identified and changed so that the theory correctly predicts the behavior. As an example, consider a part of the domain theory that describes solutions (figure 3.4) and the boiling of liquids (not shown). According to the theory, a solution boils when the boiling point of the solvent is reached.

```
Solution (?solution)
   Individuals
      ?solution
   Preconditions
      (soluble? (solute-of ?solution) (solvent-of ?solution))
   Quantity Conditions
      (greater-than (A (amount-of (solute-of ?solution))) 0)
   Relations
      (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution)))
      (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))
      (Q- (concentration ?solution) (amount-of (solvent-of ?solution)))
      (equal-to (boiling-point ?solution) (boiling-point (solvent-of ?solution)))
```

Figure 3.4 A description of a solution. The boiling point of the solution is equal to the boiling point of its solvent according to this definition.

Figure 3.5 describes a scenario in which a solution of sugar in water is heated. However, when the

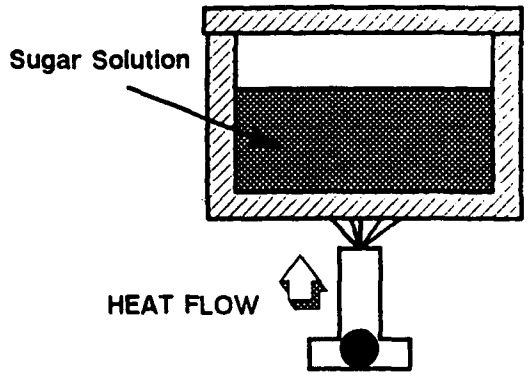


Figure 3.5 A scenario in which a solution of sugar in water is heated.

boiling point of water is reached, the sugar solution does not boil as predicted by the theory. The theory is incorrect because the boiling point of a solution is elevated due to the presence of the solute.

3.4. The Detection of Imperfect Theory Problems

The inadequacies of a domain theory are detected when it fails to correctly predict or explain the behavior for a given scenario. There are four types of failures due to imperfect theory problems:

Broken Explanation:

There are gaps in the explanation leading to an incomplete explanation. The knowledge that is required to construct the explanation is missing from the domain theory (incompleteness – type I).

Contradiction:

Explanations are constructed for conclusions that are contradictory. This problem may be due to incorrect knowledge in the domain theory (incorrectness) or due to missing knowledge (incompletenes – type I).

Multiple Explanations:

Multiple, incompatible explanations are constructed for a conclusion. This problem is due to lack of knowledge that can identify the correct explanation (incompleteness – type II).

Resources Exceeded:

The resources (time, memory, etc.) allotted are exceeded while constructing an explanation. This failure is due to an intractable theory problem.

3.5. The Detection of Incomplete and Incorrect Theory Problems in COAST

The three strategies for detecting failures due to incomplete and incorrect theories can be combined with additional assumptions to form a single, general mechanism for detecting failures due to all the above types of incomplete and incorrect theories. The general mechanism is based on contradiction detection. It involves comparing the observations with the predictions made based on the theory and generating contradictions if the two are not consistent.

The broken explanation and multiple explanations problems can be reduced to detecting contradictions if additional assumptions are made. If the closed world assumption is made – that is, the knowledge in the domain theory correctly and completely describes the behavior of the scenarios in the domain – then the broken explanation problem reduces to the contradiction problem. When a quantity is observed to change in a manner that cannot be explained by the theory, and the system, based on the closed world assumption, predicts that the quantity does not change in that manner, then this is a contradiction.

Similarly, the multiple explanations problem can also be re-expressed as the contradiction problem. If the closed world assumption is made – that is, the domain theory is complete and correct – then only consistent explanations for a conclusion are feasible. If multiple explanations can be constructed based on the theory for a conclusion and the assumptions underlying each of the explanations are incompatible then the explanations are inconsistent. For a system operating under the closed world assumption this is a contradiction.

Table 3.2 shows all the possible combinations of values for an observation and a prediction of the behavior of a particular quantity when using QP theory. There are seven different types of combinations:

1) The observed and predicted values are consistent. In this case, the theory correctly predicts the behavior of the quantity.

Observation \ Prediction	Increase	Decrease	Constant	Unknown
Increase	1	4	2	6
Decrease	4	1	2	6
Constant	3	3	1	6
Unknown	5	5	5	7

1: No Failure
2: Unexpected Observation Failure
3: Failed Prediction Failure
4: Inverse Behavior Failure
5: No Failure – Unconfirmed Predictions
6: No Failure – Observation Resolves Ambiguity
7: Ambiguity

Table 3.2 The different combinations of values for the observation and prediction of a change to a given quantity.

2) The theory predicts that the quantity remains constant but the quantity is observed to change (increase or decrease). This leads to a contradiction and this type of failure is called *unexpected observation*.

3) The theory predicts that the quantity changes (increases or decreases) but the quantity is observed to remain constant. This leads to a contradiction and this type of failure is called *failed prediction*.

4) The theory predicts that the quantity increases or decreases but the quantity is observed to decrease or increase respectively. This leads to a contradiction and this type of failure is called *inverse behavior*.

5) The theory makes a prediction for the quantity but the prediction is not confirmed because the quantity is not observed in the scenario. This does not lead to a contradiction because the real value can be consistent with the predicted value.

6) The theory cannot determine the change in the quantity. This may be due to ambiguity because of multiple opposing influences on the quantity. For example, suppose there is a flow of liquid into and out of a container. The amount of the liquid in the container can increase, decrease or remain constant depending on whether the flow into the container is greater than, less than or equal to the flow out of the container. Therefore, the theory cannot unambiguously determine the value of the change in the quantity. The observed value forces a value for the change thereby resolving the ambiguity. There is no contradiction because any of the three values that can be observed is consistent with the theory.

7) The theory cannot determine the change in the quantity and the quantity is not observed in the scenario. The value of the quantity is ambiguous. This does not lead to a contradiction because the predicted value can be consistent with the real value.

The rest of this section describes and illustrates with detailed examples the three types of failures used by COAST to detect incompleteness and incorrectness in theories – the unexpected observation failure, the failed prediction failure and the inverse behavior failure.

3.5.1. Unexpected Observation Failure

Prediction = (constant <quantity>)
Observation = (<change> <quantity>) where <change> = increase or decrease

In this type of failure, the system predicts that a particular quantity will remain constant but the quantity is observed to be changing – either increasing or decreasing.

Evaporation Example

An example involving the evaporation of liquids is used to illustrate the unexpected observation failure and the explanation formation for this type of failure. Figure 3.6 shows the scenario in which the failure is encountered.

The domain theory for this example consists of descriptions of two processes: evaporation of liquids and heat flow between physical objects (figure 3.7). The description of the evaporation

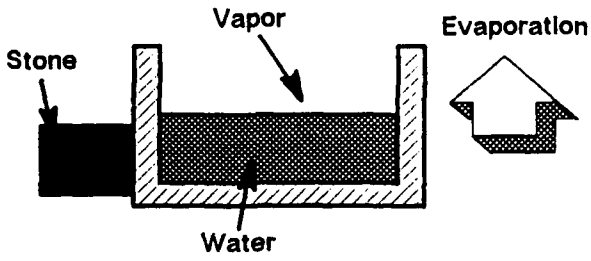


Figure 3.6 A scenario in which water is placed in an open container in contact with its vapor. A stone which is at a lower temperature than the water is in contact with the wall of the container. The wall is insulated against heat flow.

process states that when a liquid is placed in an open container its amount will decrease and the amount of the vapor will increase. These changes will take place at a rate proportional to the area of liquid in contact with the vapor. The heat flow process states that when two objects of different temperatures are connected by a heat conducting path (heat-aligned?) then the temperature of the object at higher temperature will decrease and that of the other object will increase. These changes occur at a rate that depends on the geometry of the conducting path and the temperatures of the two objects.

The scenario for the example is depicted in figure 3.6. The individuals in the layout of the scenario are the water in the container, vapor, the stone, and the two heat paths (between water and vapor and between the stone and water). The facts that are true in the scenario include: the two heat paths are not heat-aligned – that is, they do not conduct heat, the container is open, the stone is at a lower temperature than water etc.

The behavior predicted by the theory in figure 3.7 for the scenario in figure 3.6 is shown in figure 3.8. Evaporation of water is the only active process. There are two inactive processes: heat flow from water to the stone and from water to its vapor. Both heat flows are inactive because the condition for the heat flow process – heat-aligned path – is not met by the heat paths in the scenario[1]. The predicted changes are that the amount of water in the container decreases and the amount of vapor increases. All other quantities are constant.

[1]The heat path connecting the vapor and water is the common surface. The vapor is considered to be an insulating material. Therefore, though the temperatures in the layers of vapor very close to the surface are changing, the vapor as a whole is treated to be at constant temperature. The changes close to the surface are ignored.

```
Evaporation (?liquid ?vapor)
   Individuals
      ?liquid ?vapor
   Preconditions
      (open? (container ?liquid))
   Quantity Conditions
   Relations
      (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))
   Influences
      I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
      I+[(amount-of ?vapor), (A (evaporation-rate ?self))]

Heat-Flow (?source ?destination ?path)
   Individuals
      ?source ?destination ?path
   Preconditions
      (heat-aligned? ?path)
   Quantity Conditions
      (greater-than (A (temperature ?source)) (A (temperature ?destination)))
   Relations
      (Q+ (heat-flow-rate ?self) (temperature ?source))
      (Q- (heat-flow-rate ?self) (temperature ?destination))
      (Q+ (heat-flow-rate ?self) (cross-sectional-area ?path))
      (Q- (heat-flow-rate ?self) (length ?path))
   Influences
      I-[(amount-of ?source), (A (heat-flow-rate ?self))]
      I+[(amount-of ?destination), (A (heat-flow-rate ?self))]
```

Figure 3.7 *The domain theory for the evaporation example.* It consists of process definitions for evaporation and heat flow.

```
Behavior1:
   Theory: <Evaporation> <Heat-Flow>
   Scenario: <Evaporation-temperature-scenario>
   Active Processes:
            (Evaporation water vapor)
   Inactive Processes:
            (Heat-Flow water vapor vapor-path)
            (Heat-Flow water stone container-path)
   Predicted Changes:
            Increase (amount-of vapor)
            Decrease (amount-of water)
   Explanations:
      (Increase (amount-of vapor))
         I+[(amount-of vapor), (A (evaporation-rate (evaporation water vapor)))]
            (active (evaporation water vapor))
               (open? (container water))
      (decrease (amount-of water))
         I-[(amount-of water), (A (evaporation-rate (evaporation water vapor)))]
            (active (evaporation water vapor))
               (open? (container water))
```

Figure 3.8 The behavior of the evaporation example scenario predicted by the theory in figure 3.7.

The system is then given the observation:

Observation: (Decrease (temperature water)).

The system based on its domain theory makes the prediction:

Prediction: (Constant (temperature water)).

This leads to the unexpected observations failure. The system has encountered a change in a quantity when it had predicted that the quantity would remain constant.

3.5.2. Failed Prediction Failure

Prediction = (<change> <quantity>) where <change> = Increase or decrease
Observation = (constant <quantity>)

In this type of failure, the system, based on its domain theory, predicts that a particular quantity will change in a specified manner (increase or decrease) but the observations made from the real world show that, in fact, the quantity remains constant.

Dissolve Example

An example involving a substance dissolving in a solution is used to illustrate the failed prediction failure and the construction of abstract explanations for the failed prediction failure. Figure 3.9

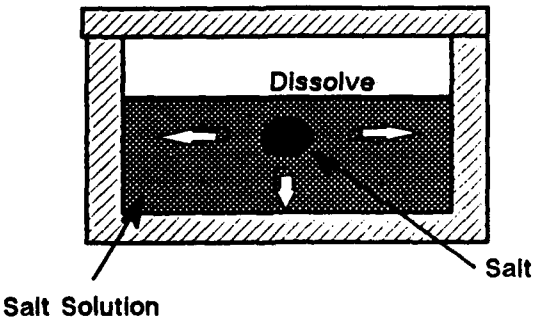


Figure 3.9 A scenario in which a solution of salt in water (brine) is placed in an open container in contact with some salt.

depicts the scenario in which the failure occurs.

Figure 3.10 shows the domain theory used for the example. The domain theory consists of a process definition for the dissolving of solids in solutions and a definition for solutions (also represented as a process but without influences). The definition for the dissolve process states that when a solid is soluble in a solution then the amount of the solid will decrease and the amount of the solute in the solution will increase. These changes will occur at a rate proportional to the area of contact between the solid and the solution. The description for solutions includes relations between various quantities that are valid when a solution exists. For example, the concentration of a solution is proportional to the amount of solute in the solution and inversely proportional to the amount of solvent in the solution. Also, in this naive description of solutions, the amount of solution is proportional to the amount of solvent and is independent of the amount of solute. In fact, this is true only if the amount of solute is small.

```
Dissolve (?solution ?solid)
   Individuals
      ?solution ?solid
   Preconditions
      (dissolves? ?solid ?solution)
   Quantity Conditions
   Relations
      (Q+ (dissolve-rate ?self) (contact-area ?solution ?solid))
   Influences
      I-[(amount-of ?solid), (A (dissolve-rate ?self))]
      I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]

Solution (?solution)
   Individuals
      ?solution
   Preconditions
      (soluble? (solute-of ?solution) (solvent-of ?solution))
   Quantity Conditions
      (greater-than (A (amount-of (solute-of ?solution))) 0)
   Relations
      (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution)))
      (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))
      (Q- (concentration ?solution) (amount-of (solvent-of ?solution)))
```

Figure 3.10 The domain theory for the dissolve example. It consists of a process definition for the dissolve process and an individual view describing a solution.

The layout of the scenario depicted in figure 3.9 includes two individuals: a solution of salt in water and salt which are placed in contact with each other in a container. The facts that are true in the scenario include: the container contains the salt solution and salt which are in contact with each other and that the salt dissolves in the salt solution.

Figure 3.11 shows the behavior predicted by the theory for the dissolve scenario shown in figure 3.9. There are two active views: the dissolving of salt in the salt solution and the salt solution. There are no inactive processes. The predicted changes include the direct effects of dissolve – an decrease in the amount of salt and an increase in the amount of solute in the salt solution. The latter change leads to a secondary change – an increase in the concentration of the salt solution. The explanations for the predicted changes are also shown in the behavior.

```
Behavior1:
   Theory: <Dissolve> <Solution>
   Scenario: <Dissolve-saturation-scenario>
   Active Processes: (Dissolve salt-solution salt) (solution salt-solution)
   Inactive Processes:
   Predicted Changes:
      (Decrease (amount-of salt))
      (Increase (amount-of (solute-of salt-solution)))
      (Increase (concentration salt-solution))
   Explanations:
      (Decrease (amount-of salt))
         I-[(amount-of salt),
               (A (Dissolve-rate (Dissolve salt-solution salt)))]
            Active (Dissolve salt-solution salt)
               (Dissolves? salt salt-solution)
      (Increase (amount-of (solute-of salt-solution)))
         I+[(amount-of (solute-of salt-solution)),
               (A (Dissolve-rate (Dissolve salt-solution salt)))]
            Active (Dissolve salt-solution salt)
               (Dissolves? salt salt-solution)
      (Increase (concentration salt-solution))
         (Q+ (concentration salt-solution) (amount-of (solute-of salt-solution)))
            (Active (solution salt-solution))
               (Greater-than (A (amount-of (solute-of salt-solution))) 0)
               (soluble? (solute-of salt-solution) (solvent-of salt-solution))
         (Increase (amount-of (solute-of salt-solution)))
            I+[(amount-of (solute-of salt-solution)),
                  (A (Dissolve-rate (Dissolve salt-solution salt)))]
               Active (Dissolve salt-solution salt)
                  (Dissolves? salt salt-solution)
```

Figure 3.11 The behavior for the dissolve example scenario based on the domain theory described in figure 3.10.

The system is given the observation:

```
Observation: (constant (amount-of salt)).
```

However, the system, based on its domain theory, makes the following prediction for the given scenario:

Prediction: (decrease (amount-of salt)).

This results in a failed prediction failure because the system's predicted decrease in the amount of the salt in the container is not observed in the scenario.

3.5.3. Inverse Behavior Failure

Prediction = (<change> <quantity>) where <change> = increase or decrease
Observation = (<opposite-change> <quantity>)
where <opposite-change> = decrease or increase

In this type of failure, the system predicts that a particular quantity will increase or decrease. However, it is observed that the opposite change occurs, that is, the quantity decreases or increases, respectively.

Flow Example

An example involving the flow of a liquid between two containers connected by a pipe is used to illustrate the inverse behavior failure and the explanation construction for this type of failure. Figure 3.12 shows the scenario. The domain theory for the example consists of only one process –

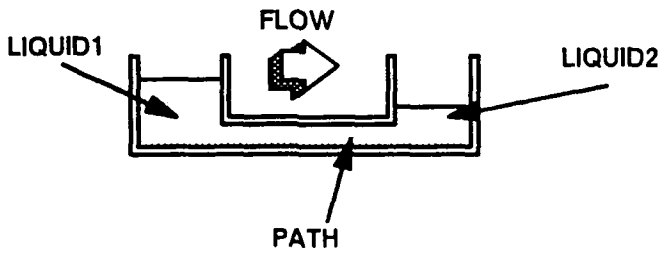


Figure 3.12 A scenario in which liquid1 is connected to liquid2 by a path. The pressure at liquid1 is greater than the pressure at liquid2. The path permits the flow of liquid.

a definition for the flow of liquids from a source to a destination through a path. The domain theory is described in figure 3.13.

```
Liquid Flow (?source ?destination ?path)
   Individuals
      ?source ?destination ?path
   Preconditions
      (fluid-flow-aligned? ?path)
   Quantity Conditions
      (greater-than (A (pressure ?source)) (A (pressure ?destination)))
   Relations
      (Q+ (flow-rate ?self) (pressure ?source))
      (Q- (flow-rate ?self) (pressure ?destination))
      (Q+ (flow-rate ?self) (cross-sectional-area ?path))
      (Q- (flow-rate ?self) (length ?path))
   Influences
      I-[(amount-of ?source), (A (flow-rate ?self))]
      I-[(amount-of ?destination), (A (flow-rate ?self))]
```

Figure 3.13 The domain theory for the flow example. It consists of only one process definition – the definition for the flow of liquids from a source to a destination through a path.

The flow process requires two conditions: the path should be flow-aligned? – that is, it should permit the flow of liquid (for example, all the valves should be open) and the pressure at the source should be greater than the pressure at the destination. When these conditions are met, the flow process predicts that the amount of liquid at the source will decrease and the amount of liquid at the destination will also decrease[2]. These changes will occur at a rate that depends on the geometry of the path and the pressures at the source and destination.

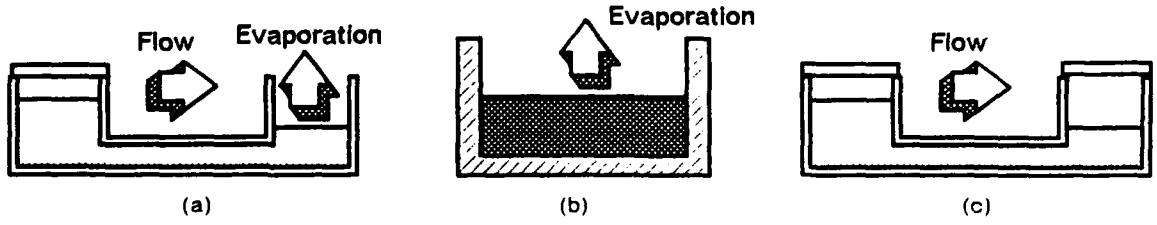


Figure 3.14 (a) A scenario in which alcohol in one container is connected to alcohol in a second, open container by a hollow pipe. (b) A scenario in which water is placed in an open container. (c) A scenario in which water in one container is connected to water in a second, closed container.

[2] At first glance, this example may seem rather artificial and contrived. However, the reader should note that such incorrect theories are quite frequently learned by theory revision systems such as the one described in the thesis. Suppose the system initially does not have any knowledge about the flow of liquids or evaporation and it encounters the scenario shown in figure 3.14a. In this scenario, alcohol in the first container is connected to alcohol in the second, open container through a hollow pipe. The amounts of alcohol in both the containers is observed to be decreasing. Based on this example, the theory revision system generates a new process that involves a flow of alcohol and incorrectly describes the acquired flow process to cause a decrease in the amounts of the source and destination liquids. Since such incorrect theories are frequent it is important that the theory revision system have the ability to recover from these mistakes when further examples are provided. The system described in this thesis has the ability to recover from such incorrect theories. For example, when the scenario shown in figure 3.14b is encountered in which the amount of water in an open container decreases, it can formulate a new process that can explain this observation. Then, when it encounters the scenario shown in figure 3.14c in which water flows from one, closed container to another, it can correct the previously learned flow process. Importantly, the system can ensure that the revised theory explains the observations made in all three scenarios.

The layout of the scenario depicted in figure 3.12 consists of three individuals – liquid1 in the first container, liquid2 in the second container and the path connecting the two containers. Some of the facts that are true in the scenario are: the path is flow–aligned? and the pressure at container1 is greater than the pressure at container2.

The behavior predicted for the scenario based on the theory is shown in figure 3.15. The only active process is the flow of liquid1 to liquid2 through the path. The changes predicted by the theory are that the amounts of liquid1 and liquid2 will decrease.

```
Behavior1:
   Theory: <Liquid–Flow>
   Scenario: <Liquid–Flow–Scenario>
   Active Processes: (Liquid–Flow liquid1 liquid2)
   Inactive Processes:
   Predicted Changes:
      (Decrease (amount–of liquid1))
      (Decrease (amount–of liquid2))
   Explanations:
      (Decrease (amount–of liquid1))
         I–[(amount–of liquid1), (A (Flow–rate (Flow liquid1 liquid2 path)))]
            Active (Flow liquid1 liquid2 path)
               (Fluid–flow–aligned? path)
               (greater–than (A (pressure liquid1)) (A (pressure liquid2)))
      (Decrease (amount–of (solute–of salt–solution)))
         I–[(amount–of liquid2), (A (Flow–rate (Flow liquid1 liquid2 path)))]
            Active (Flow ilquid1 liquid2 path)
               (Fluid–flow–aligned? path)
               (greater–than (A (pressure liquid1)) (A (pressure liquid2)))
```

Figure 3.15 The behavior predicted by the theory described in figure 3.13 for the scenario shown in figure 3.12.

Based on the theory described in figure 3.13 the system makes the following prediction about the given scenario:

Prediction: (decrease (amount liquid2)).

However, the observed change is:

Observation: (increase (amount liquid2)).

The observed change is directly opposite to the predicted change resulting in the inverse behavior failure.

3.6. Discussion

This chapter has described a taxonomy of imperfect theory problems and different techniques for detecting these problems. The taxonomy is based on extensions and revisions to a classification of imperfect theory problems in the context of explanation–based generalization by Mitchell et al. [Mitchell86]. Rosenbloom and Laird [Rosenbloom86] describe two additional categories to the three categories listed by Mitchell et al. – errors in the domain theory and a defeasible domain theory. These two imperfections are included in the incorrect theories and the contradiction mechanism can be used to detect these types of problems.

Many different strategies have been used to detect problems with the domain theory. The failure of a plan to achieve its goals is employed by Doyle [Doyle86] and Chien [Chien87] to detect problems due to approximations to a complete theory to make it tractable. Likewise, Carbonell and Gill [Carbonell87] use plan execution failures to detect incompleteness in theories. The failure to explain an example or observation is used by Hall [Hall86] and Falkenhainer [Falkenhainer87a] to detect problems with the theory. Dietterich and Flann [Dietterich88] use multiple explanations to detect incomplete theories. Pazzani [Pazzani88a] uses contradictions to detect incorrect theories.

This chapter has described a taxonomy of imperfect theory problems. Four strategies for detecting problems with the domain theory were described. A generalized version of contradiction detection was shown to be adequate to detect incompleteness and incorrectness in domain theories. Finally, a number of examples from QP theory were described to illustrate the different failures due to incomplete and incorrect theories.

CHAPTER 4

HYPOTHESIS GENERATION FOR THEORY REVISION

4.1. Constraints for Hypothesis Generation

Hypothesis generation for theory revision is a complex problem. It involves identifying the inadequacies of the existing theory and making those revisions that will solve the problems encountered by the theory for the given scenario. In general, for a non-trivial domain theory, a very large number of hypothesized revisions is feasible. This poses a problem because it will be virtually impossible to test each revised theory. Therefore it is important to exploit all the available sources of knowledge to restrict the size of the hypothesis space. The remainder of this section shows how knowledge about the failure, the scenario in which the failure occurred, the explanation construction process and the structure of the hypothesis space can be used to constrain a hypothesis generator for theory revision. The following sections describe how such a hypothesis generator is implemented in COAST for domain theories represented in Qualitative Process theory.

4.1.1. Theory Revision Operators

The basic requirement of a hypothesis generator for theory revision is the ability to modify the existing theory. This requirement may be met by equipping the generator with *theory revision operators* – operators that can be applied on the existing theory to produce revised theories.

Let the representation language for the domain theory be the language L. Given any initial domain theory Ti from the language L and the set Sf of theories in L that can correctly predict the behavior of any scenario S from a language of scenarios then a set of theory revision operators O is defined to be *complete* if at least one member of set Sf can be generated from Ti using the operators O. The set of theory revision operators is defined to be *correct* if only legal theories (based on the syntax of L) can be constructed by applying the theory revision operators on any initial theory in L.

A hypothesis generator for theory revision must have a complete and correct set of theory revision operators. If the set is incomplete then the generator may not be able to propose a theory that correctly predicts the observed behavior. For example, suppose the set of theory revision operators lacks an operator to delete components of a theory. Then, if the initial theory has an incorrect component and if the behavior of a scenario can be correctly predicted only if the incorrect component is deleted, then the hypothesis generator will never be able to a generate a theory that will correctly predict the behavior of that scenario. If the set is not correct, then the hypothesis generator may produce theories that are not in L, and which can not be further revised since the operators will no longer be applicable.

An operational hypothesis generator for theory revision can be constructed based solely on a complete and correct set of theory revision operators. Figure 4.1 shows such a basic hypothesis

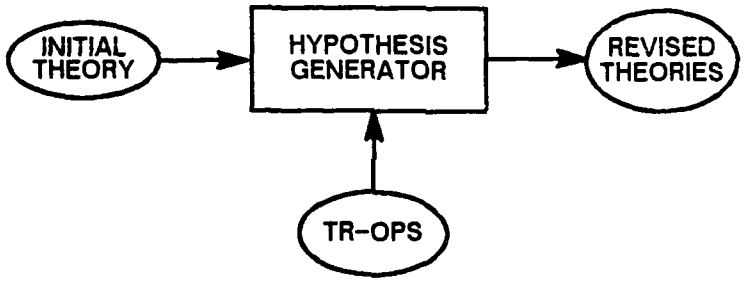


Figure 4.1 A basic hypothesis generator for theory revision.

generator. It accepts a theory as input and produces revised theories as output. The revised theories are produced based on applications of the theory revision operators on the initial theory. In practice, the space of generated theories, S (figure 4.2), for an initial theory can be horrendously large. Suppose the domain theories are represented in QP theory. A simple domain might consist of 10 processes with an average of 10 components each (individuals, preconditions, quantity conditions, relations, influences and the process itself). Let the set of theory revision operators consist of 5 operators (for example, adding a new component, deleting a component, inverting a component, widening the scope of a component and narrowing the scope of a component). Suppose, on an average, the application of an operator produces 10 revised theories (for example, the number of new quantity conditions that can be added to a theory to produce new theories

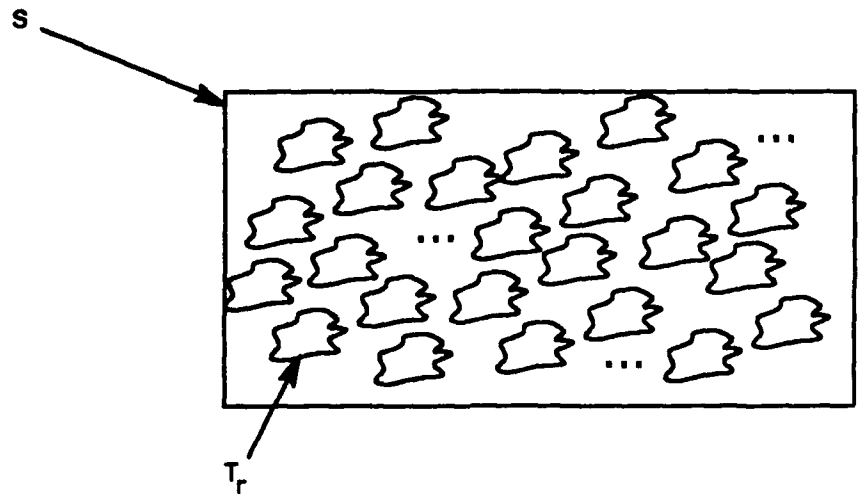


Figure 4.2 The space of revised theories.

depends on the number of quantities in the theory, say 20, while deleting a quantity condition will produce 1 revised theory). The total number of revised theories (based on one application of one theory revision operator) is:

$$\text{No. of revised theories} = \text{No. of processes} * \text{No. of components per process} * \text{No. of operators} * \text{Avg. yield of an operator} = 10 * 10 * 5 * 10 = 5000$$

In practice, it is impossible to test all these revised theories to determine which theory correctly predicts the observed behavior. Therefore, further constraints will have to be imposed on the hypothesis generator to restrict the revised theories to a manageable number.

4.1.2. Scenario Constraint

The first constraint that is imposed on hypothesis generation is the context in which the failure is encountered – the failure scenario. Under this constraint the hypothesis generator will generate only those theories that are relevant to the scenario in which the failure is detected. Those components of the theory that were accessed to predict the behavior of the scenario are considered relevant to the hypothesis generation of revised theories and those components of the theory that were not accessed are ignored. Figure 4.3 shows the hypothesis generator which uses the scenario constraint. Revised theories are generated by applying the set of theory revision operators to those

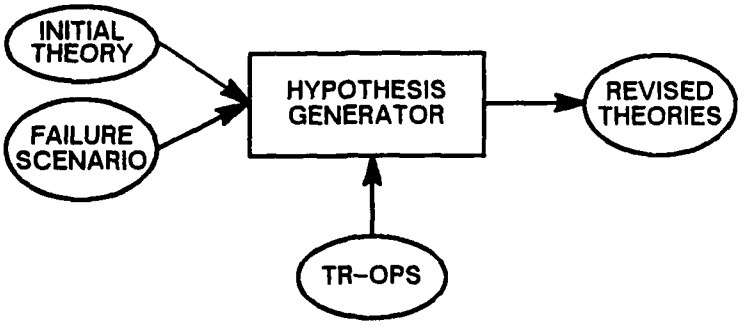


Figure 4.3 A hypothesis generator that incorporates the scenario constraint.

components of the theory that are relevant to the failure scenario. Figure 4.4 shows the space of

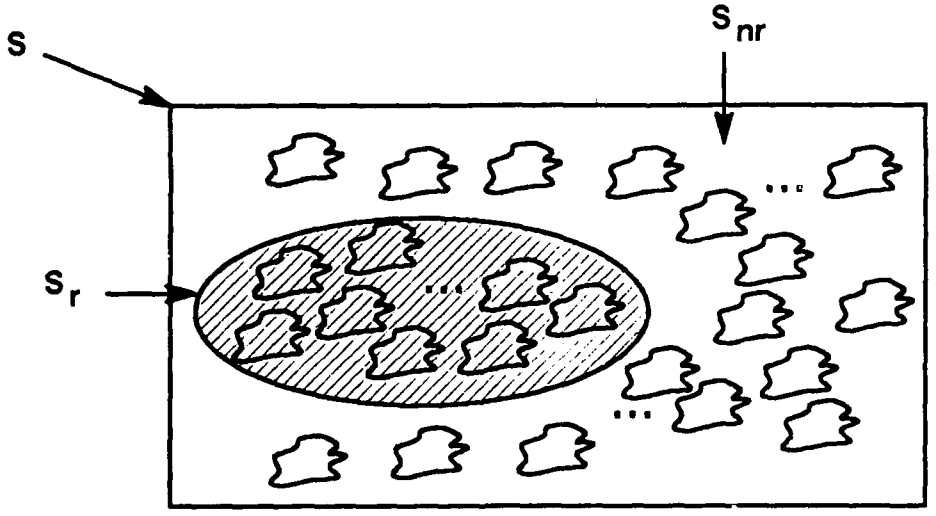


Figure 4.4 The space of revised theories. Sr is the space of revised theories that are relevant in the context of the failure scenario.

revised theories, Sr, produced by the hypothesis generator.

Introducing the scenario constraint may lead to the selection of a revised theory that is not optimum in terms of the simplicity and size of the theories. For example, simpler theories that correctly predict the behavior of the scenario may involve modifications to components deemed irrelevant by the scenario constraint since they were not accessed. Note that the hypothesis generator will still

produce theories that will correctly predict the behavior of the scenario. However, these theories may involve adding new components to the theory.

The scenario constraint is a powerful method for limiting the number of hypotheses produced. In the example introduced in the previous section, the initial theory consists of 10 processes. If only 3 of these are considered during the prediction of the behavior of the given scenario as being active or with the potential to become active (if the conditions become suitable) then the number of theories produced are:

No. of theories produced = No. of processes relevant to the scenario *
No. of components * No. of operators *
Avg yield of operators
= 3 * 10 * 5 * 10
= 1500

However, the number of theories produced can still be too large to test and further constraints have to be incorporated into the generation process.

4.1.3. Explanation Constraint

An additional source of constraint for hypothesis generation is the failure that triggered theory revision. Under this constraint only those hypotheses that produce theories that can successfully explain the observation that led to the failure are generated. A hypothesis generator that incorporates the explanation constraint has to be equipped with knowledge about the explanation construction process. In particular, it must be able to categorize failures, analyze each category of failure to determine what type of explanations are feasible, and construct those explanations, based on the hypothesized revisions to the theory, for the given failure and scenario.

Figure 4.5 shows the architecture of a hypothesis generator that incorporates the explanation constraint and the scenario constraint. The hypothesis generator examines the failure and constructs explanations for the observations that led to the failure based on hypothesized revisions to those components of the theory that are relevant according to the scenario constraint. Figure 4.6 shows the relationship between the different spaces of proposed theories. Se is the space of revised theories produced by the hypothesis generator.

The hypothesis generator incorporating the explanation constraint is complete if: 1) all the encountered failures are classifiable into one of the known failure categories, and 2) it is possible to

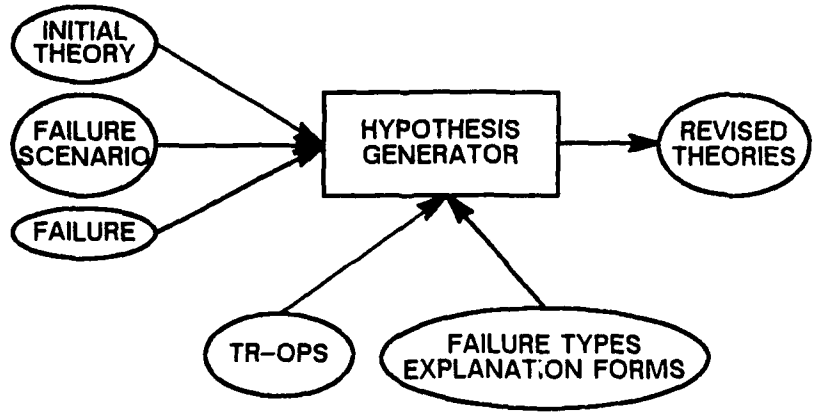


Figure 4.5 A hypothesis generator incorporating the scenario and explanation constraints.

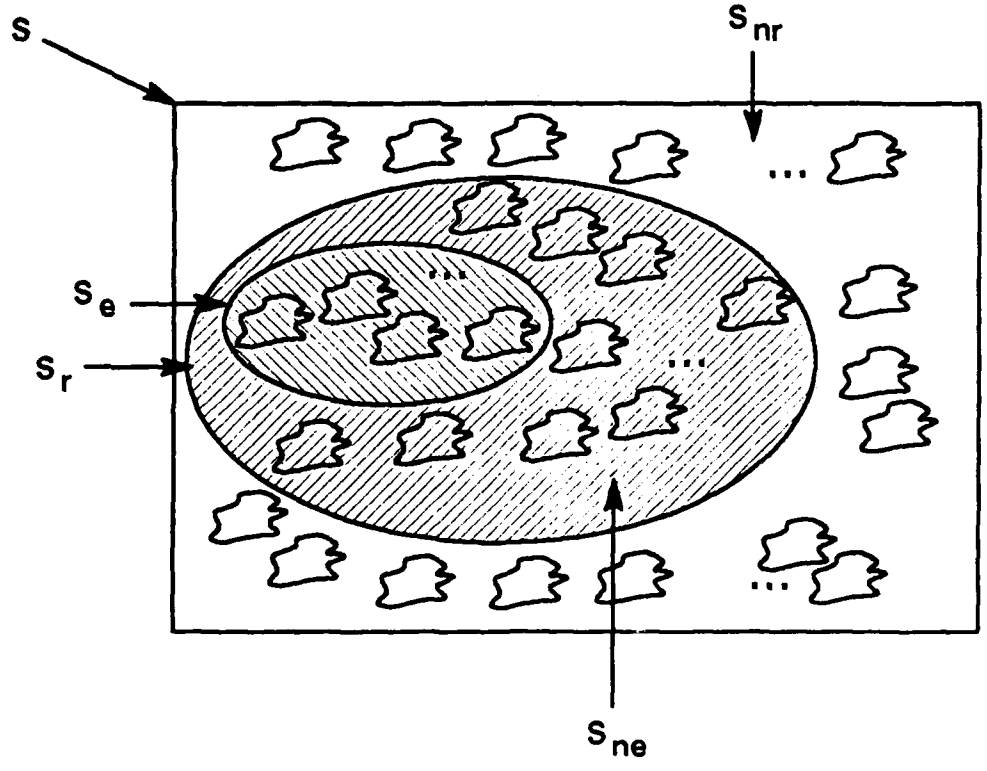


Figure 4.6 The space of revised theories. Se is the space of theories that can explain the observed changes that lead to the failure.

generate all the different types of explanations for each failure based on hypothesized revisions to the theory.

The explanation constraint is also powerful in limiting the number of hypotheses produced. It restricts the type of components modified and the types of operators that are applicable. Suppose, the failure is due to an observed change which is not predicted by the initial theory. If there is only one active process in the scenario, then only the effects of that process have to be revised. Modifying the conditions of the active process will not explain the observed change. Also, only revisions such as adding an effect or modifying an existing effect can explain the observed behavior. Deleting an effect from the active process will not explain the observed change. In this manner, both the components that can be revised and the types of revision on these components are constrained. In the example of the previous section, suppose the number of components that can be modified is 5 and the number of operators that are applicable is 3, then the number of revised theories generated is:

No. of theories produced = No. of processes relevant to the scenario
* No. of components * No. of operators
* Avg yield of operators
= 3 * 5 * 3 * 10
= 450

4.1.4. Abstraction Constraint

Even after the scenario and explanation constraints are incorporated into the hypothesis generator, the number of theories that can be proposed can be very large. Another source for constraining the number of hypotheses generated is *abstraction*. Under this constraint a group of hypotheses are clustered together into an abstract hypothesis. The clustering has the following property: if the abstract hypothesis is tested and found to be false then none of the hypotheses in its cluster is true. The hypothesis generator first produces a set of abstract hypotheses. These hypotheses are tested. The abstract hypotheses that are successful are refined to produce the refined hypotheses. There can be more than one level of abstraction and a number of refinement and testing steps may be required before the hypothesis generator proposes *concrete* hypotheses that correspond to revised theories.

Figure 4.7 shows the hypothesis generator incorporating abstraction and refinement. It accepts an additional input: a set of abstract hypotheses which are to be refined. The hypothesis generator

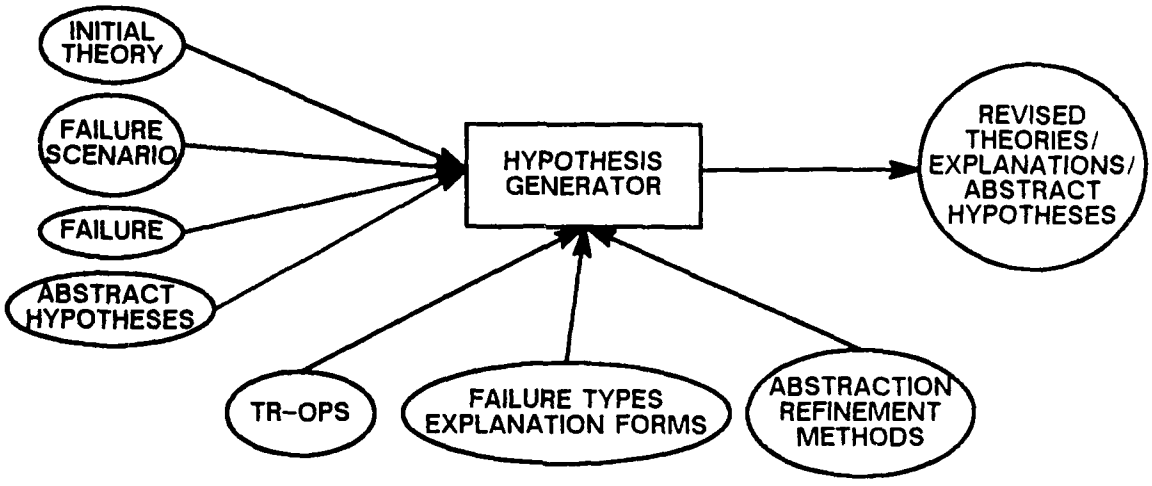


Figure 4.7 A hypothesis generator incorporating the scenario, explanation and abstraction constraints.

functions in two modes: 1) If it is being called without any abstract hypotheses then it proposes abstract hypotheses and constructs explanations for the observation based on abstract hypotheses. 2) If it is being called with abstract hypotheses, it returns the refined hypotheses and explanations based on the refined hypotheses. If the refined hypotheses are hypotheses that propose revisions to the initial theory then it returns the revised theories. Figure 4.8 shows the relationship among the different spaces of proposed theories. The space of abstract hypotheses, Sh, corresponds to a subset of the space of theories that explain the observed change Se.

The abstraction constraint is effective if: 1) The representation language for the domain is such that large clusters of hypotheses are possible for each abstract hypothesis. Then if an abstract hypothesis is refuted in the testing stage none of the hypotheses in its corresponding cluster will have to be generated or tested. 2) The abstract hypotheses can be tested without examining the hypotheses in its corresponding cluster.

Domain theories represented in QP theory satisfy the above conditions, thereby, permitting the effective use of abstraction spaces for hypothesized theories. For example, suppose the hypothesis generator makes an assumption that either the conditions of a process or its effects can be revised but not both. This is a locality of fault assumption based on the functionality of the processes. A violation of the assumption implies that the process did not activate correctly and it

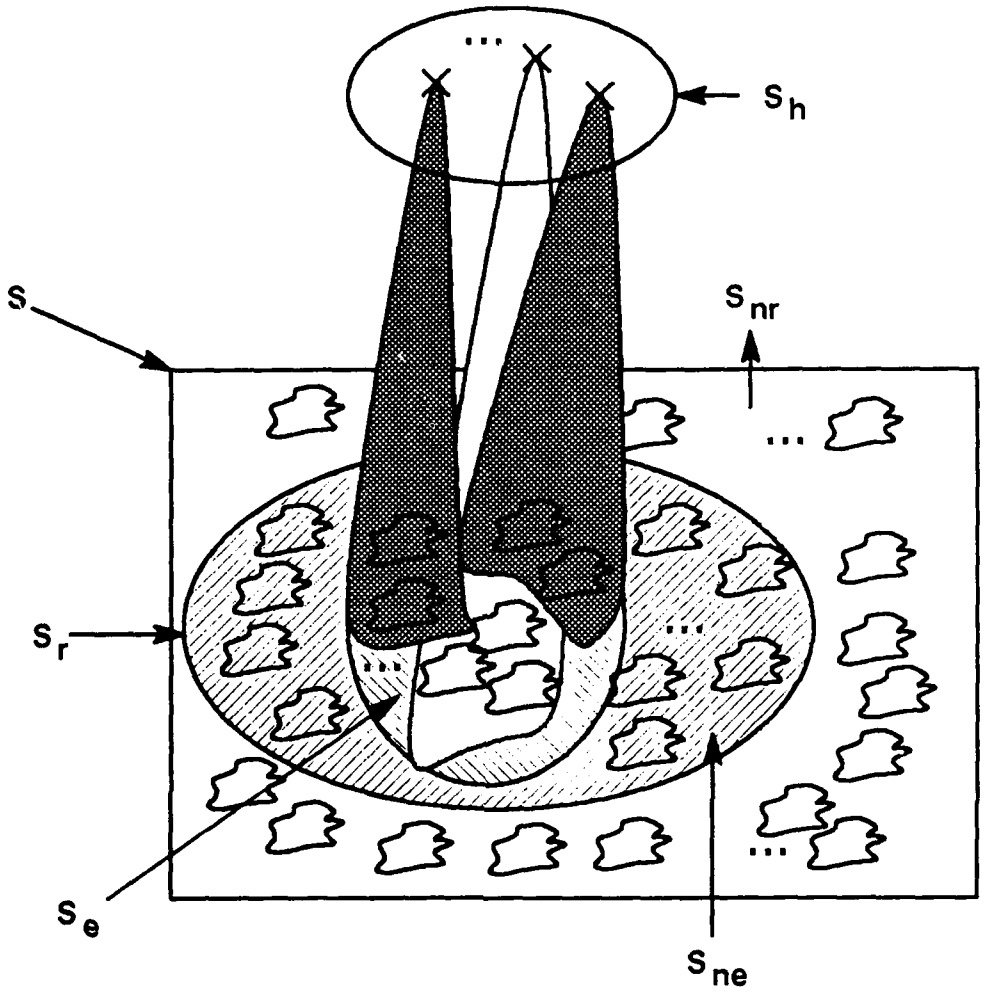


Figure 4.8 The space of revised theories S. The hypothesis generator of figure 4.7 generates abstract hypotheses in space Sh and refines them to the theories in space Se.

did not produce the correct effects. Note that this is different from the single fault assumption because a number of revisions can be made simultaneously to the conditions or the effects. If the locality of fault assumption is made, then the theory revision hypotheses can be clustered into abstract hypotheses that question whether a process is active or not (conditions to be revised) or whether a process causes a change or not (effects to be revised). These abstract hypotheses can be tested without examining the different types of revised theories. For example, to test whether a process is active, the system can check if all the effects of the process (which, by the locality of fault assumption, are correct) are true in the scenario.

The abstraction constraint has considerable potential for constraining the number of hypotheses generated. In the example of the previous section, suppose each of the three processes examined yield 3 abstract hypotheses (each process has three instantiations in the scenario) and during the testing stage seven of these hypotheses are eliminated. Then the number of hypotheses generated is:

No. of hypotheses generated = no. of abstract hypotheses generated
+ no. of refined hypotheses for valid abstract hypotheses
= 3 * 3 + 2 * 5 * 3 * 3
= 59

4.1.5. Building an Effective Hypothesis Generator

To summarize, in order to build a hypothesis generator that proposes a small number of revised theories, the requirements are:

1) It must possess a complete and correct set of theory revision operators.

2) It must have access to the behavior predicted by the initial theory for the failure scenario. This will allow it to determine those components of the theory that are relevant in the context of the scenario in which the failure occurred.

3) It must have knowledge about the different types of failures and the types of explanations feasible for each type of failure and the hypothesized revisions under which the explanations can be constructed.

4) It must have knowledge about the abstract hypotheses feasible for each type of failure and the knowledge to refine each abstract hypothesis.

In restricting the number of hypotheses by applying the constraints, the hypothesis generator may have made assumptions about the types of theories that can be generated – for example, the locality of fault assumption implies that a new process is preferred to fixing an existing process by revising both its conditions and effects. While these assumptions limit the type of theories that can be generated; they do not make the hypothesis generator incomplete. The hypothesis generator will always propose a revised theory that will explain the observation that led to the failure though the revisions can degenerate into proposing a new process for every observation that triggers a failure.

4.2. Hypothesis Generation in COAST

Based on the above discussion, a hypothesis generator for the theory revision of domains represented in QP theory is described. Figure 4.9 shows the architecture for the hypothesis

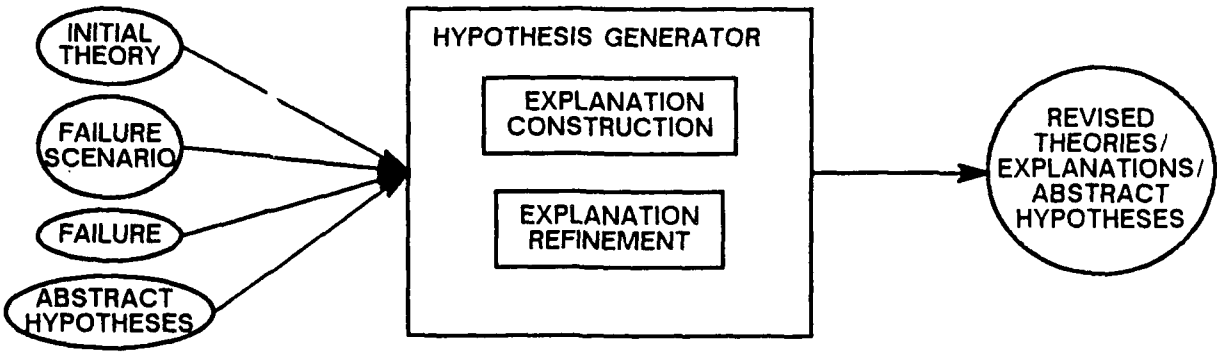


Figure 4.9 A hypothesis generator for theories represented in QP theory. It incorporates the constraints due to the scenario, explanation and failure.

generator. The inputs to the hypothesis generator are the initial theory that has to be revised, the failure that triggered the theory revision, the scenario in which the failure is encountered and the behavior predicted by the initial theory for the scenario. Based on these four inputs, the hypothesis generator produces hypotheses for making revisions to the theory, explanations for the failure based on these hypotheses and the revised theories.

The top level procedure for revising theories is shown in figure 4.10. The hypothesis generator constructs explanations based on abstract hypotheses for the observation that led to the failure. These hypotheses are tested and the remaining hypotheses are input to the hypothesis generator. The hypothesis generator refines these hypotheses and constructs explanations based on the refined hypotheses. These are again tested and the remaining hypotheses are again refined. This continues until the refined hypotheses correspond to revisions to the domain theory. In this case, the hypothesis generator also constructs the revised theories based on these hypotheses The hypothesis generator applies the scenario, explanation and abstraction constraints while generating hypotheses. The next three sections elaborate on the explanation construction and refinement procedures. The next section describes the different types of abstract hypotheses used by the system. The following section describes how these abstract hypotheses are used in constructing

explanations for each type of failure. The next two sections describe how the abstract hypotheses are refined and how explanations and revised theories are constructed based on the refined hypotheses.

```
Procedure Hypothesis-Generation (theory scenario behavior failure abstract-hypotheses)
  If (null abstract-hypotheses)
     then
        Explanation-Construction (theory scenario behavior failure)
        ;;; Returns explanations and abstract hypotheses.
     else
        unless no more refinement do
           Explanation-Refinement (theory scenario behavior failure abstract-hypotheses)
           ;;; Returns explanations, refined hypotheses and revised theories if applicable.
```

Figure 4.10 The procedure for generating hypotheses.

4.3. Abstract Hypotheses

As explained in section 4.1.4, the number of proposed theories can be considerably restricted if it is possible to formulate abstract hypotheses that cover a large number of revisions to the theory. For domain theories represented in QP theory, this type of structuring of the hypothesis space is feasible. Abstract hypotheses are proposed which question whether a process is functioning properly or whether a quantity is changing. The hypotheses at this level are abstract because they do not propose concrete changes to the domain theory – it is not possible to construct the revised theories corresponding to the hypothesis without examining the refined hypotheses.

For example, suppose a change is observed that is not predicted by the theory. Suppose there is a process that has been predicted to be inactive in the given scenario but which could explain the observed change if it were active. An abstract hypothesis suggesting that the inactive process is active can be formulated. This abstract hypothesis is refined by examining each failed condition of the process and proposing a revised condition that is satisfied in the given scenario. If there are many conditions a large number of hypotheses will be generated. Without abstraction, each of these hypotheses will have to be tested. With abstraction, the abstract hypothesis is first tested and if it can be eliminated then it is not necessary to test any of the refined hypotheses. Abstraction is effective only if it is possible to test the abstract hypothesis without examining the refined hypotheses. In the case of the abstract hypothesis proposing that the inactive process is active, if the hypothesis is true then all the effects of the process must hold in the scenario. Therefore the hypothesis can be tested by conducting experiments to check for the net consequences of each

effect of the process (in conjunction with the effects of the other active processes in the scenario). If the experimental observations are not compatible with the computed net consequences of the process which is hypothesized to be active then the process is not active and the abstract hypothesis can be eliminated.

The advantages of using abstract hypotheses are:

1) It is possible to localize the problem with the domain theory to a component of a process such as the failed conditions of a process in the above example. This is of considerable help when the theory is complex and consists of a large set of processes.

2) In general, a large number of revision operators may be applicable to a faulty component. The abstract hypothesis and the construction of explanations for the observed behavior considerably constrain the set of revision operators that can be applied to the faulty components of the theory. In the above example, adding new conditions to the inactive process is not useful because the added conditions will not make the inactive process active. Therefore this type of revisions do not conform with the abstract hypothesis.

Eight types of abstract hypotheses are required to construct explanations for the three types of failures that were described in chapter 3:

a) Active?

(Active? ?process)

The process *?process* is hypothesized to be active. The process selected for such a hypothesis is a process that is predicted to be inactive in the scenario based on the initial domain theory. The hypothesis suggests that the conditions of the selected process are wrong but it does not describe why the process is active – that is, how the existing conditions have to be revised in order for the process to become active.

b) Causes?

(Causes? ?process ?observation)

The process *?process* is hypothesized to cause the observation *?observation*. The process selected is one that is predicted to be active in the given scenario by the initial domain theory. The

hypothesis suggests that the effects of the process are wrong. The hypothesis does not specify how the effects have to be revised in order for the process to cause the observation.

c) Inactive?

(Inactive? ?process)

The process *?process* is hypothesized to be inactive. The process selected for the hypothesis is one that is predicted to be active in the given scenario. This hypothesis suggests that conditions of the process are wrong but it does not describe how the conditions can be revised to make the process inactive.

d) Not-causes?

(Not-causes? ?process ?observation)

This hypothesis states that *?process* does not cause the change described by the observation *?observation*. The process selected for the hypothesis is a process that is predicted to be active in the given scenario and is responsible for causing the observed change. The hypothesis suggests that the effects of the process may be wrong. It does not specify how the effects can be changed to prevent predicting the observed change for the scenario.

e) Unexpected-observation?

(Unexpected-observation? ?change)

A change *?change* is hypothesized. This hypothesis may be made when a change is predicted and either no change or the reverse change is observed. To explain such failures this hypothesis is used along with other hypotheses such as Equals? or Dominates?. The change selected for the hypothesis is a change that opposes the predicted change. It does not specify how the domain theory has to be revised to predict the opposing change that has been hypothesized.

f) Equals?

(Equals? ?change1 ?change2)

This hypothesis states that two changes are equal. It is made in conjunction with the *unexpected-observation?* hypothesis. One of the changes may be a predicted change and the other

may be a hypothesized change that nullifies the predicted change thereby accounting for the failure to observe the predicted change.

g) Dominates?

(Dominates? ?change1 ?change2)

?Change1 is hypothesized to dominate *?change2*. This hypothesis is made in conjunction with the *unexpected-observation?* hypothesis. Suppose the initial theory predicts *?change2* but *?change1* – a change opposite to *?change2* – is observed. Then a change *?change1* can be hypothesized by the *unexpected-observation?* hypothesis and can be hypothesized to dominate *?change2* thereby accounting for the observed inverse behavior.

h) New-process?

(New-process? ?process)

A new process *?process* is hypothesized. The new process has conditions such that it is active in the given scenario and effects such that it causes the observed phenomenon. However, the actual components of the new process are not described by the abstract hypothesis.

4.4. Explanation Construction based on Abstract Hypotheses

Explanations for the observations in each of the failures – unexpected observation, failed prediction and inverse behavior – can be constructed based on the abstract hypotheses described in the above section. Each failure type is analyzed and the different types of explanations for the observation that led to the failure are determined. These types of explanations are independent of the domain or the scenario and depend only on the language of representation – QP theory. The procedure for constructing explanations is shown in figure 4.11. The method for constructing the explanations is:

1) Classify the given failure into one of the three failure categories.

2) For each category, the knowledge about the types of explanations that can be constructed is used to construct explanations for the given failure based on the abstract hypotheses described above.

```
Procedure Explanation-Construction (theory scenario behavior failure)
   Case (failure-type failure)
      Unexpected-observation
         (Unexpected-observation-explanations theory scenario behavior failure)
      Failed-Prediction
         (Failed-prediction-explanations theory scenario behavior failure)
      Inverse-behavior
         (Inverse-behavior-explanations theory scenario behavior failure)
```

Figure 4.11 *The procedure for forming explanations based on abstract hypotheses for each type of failure.*

There are three types of explanations corresponding to each type of failure – 1) unexpected observation explanations 2) failed prediction explanations and 3) inverse behavior explanations.

4.4.1. Explanation Construction for Unexpected Observation Failures

There are three types of explanations for this failure – causes? explanations, active? explanations and new-process? explanations. The procedure for constructing the explanations is shown in figure 4.12.

```
Procedure Unexpected-observation-explanations (theory scenario behavior failure)
   Collect explanations from:
      (causes?-explanations theory scenario behavior failure)
      (active?-explanations theory scenario behavior failure)
      (new-process?-explanations theory scenario behavior failure)
```

Figure 4.12 The procedure for constructing explanations for the unexpected observation failures.

4.4.1.1. Causes? Explanations

This type of explanation is based on the abstract hypothesis *causes?*. An active process is hypothesized to cause the observation. The initial theory fails to predict the change because the process is not correctly represented – in particular, its effects i.e its relations or influences are incorrect. The general form of the explanation resulting from this hypothesis is shown in figure 4.13. The procedure for constructing such explanations is described in figure 4.14. A candidate set of active processes for the hypotheses is generated using the set of active processes in the scenario. If there are too many active processes heuristics can be used to prune them. For example, processes that affect the object involved in the quantity can be preferred to those that do not.

(<change> <quantity>)
  (Causes? ?active-process (<change> <quantity>))
    H: Flawed effects ?active-process
  (Active ?active-process)
    <Explanation for conditions>

Figure 4.13 The general form of the causes? explanations.

Procedure Causes?-Explanations (theory scenario behavior failure)
  Generate a set of candidate processes from the set of active processes
  For each process in the candidate set
    do
      Hypothesize that the process causes the observed change
      Construct the abstract explanation based on this hypothesis and the form of the explanation in figure 4.13
  Return the explanations and the hypotheses

Figure 4.14 The procedure for constructing the causes? explanations.

Example

Consider the evaporation example described in section 3.5.1. In this example an unexpected decrease in the temperature of water is observed in the scenario (reproduced in figure 4.15). The

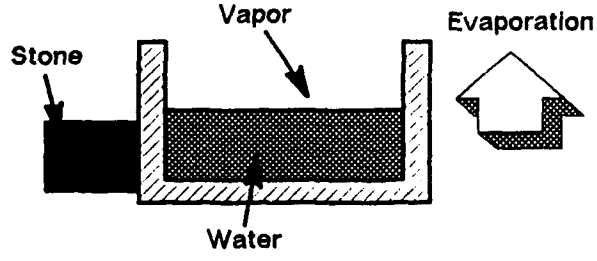


Figure 4.15 The scenario for the evaporation example.

only active process is the evaporation of water. This process is hypothesized to cause the decrease in the temperature of water. The explanation for this change based on this hypothesis and constructed according to the explanation form of figure 4.13 is shown in figure 4.16.

(decrease (temperature water))
(Causes? (evaporation water vapor) (decrease (temperature water))
H: Flawed effects (evaporation water vapor)
(Active (evaporation water vapor))
(open? (container water))

Figure 4.16 The explanation for the observed decrease in the temperature of water based on the hypothesis that evaporation is causing the decrease.

4.4.1.2. Active? Explanations

This type of explanation is based on the abstract hypothesis *active?*. An inactive process is hypothesized to be active. The process selected for the hypothesis is such that if it is active in the scenario it can explain the observed change. Therefore, the problem with the initial theory, according to this hypothesis, is that the conditions of the process are incorrect and need to be revised so that the process is active in the failure scenario. The general form of the explanation constructed based on this abstract hypothesis is shown in figure 4.17. The procedure for constructing such explanations for an observed change is shown in figure 4.18. The candidate set for selecting processes for the abstract hypothesis is formed by selecting only those processes, from the set of inactive processes in the scenario, that can each cause the observed change if it is made active.

(<change> <quantity>)
(Causes ?inactive-process (<change> <quantity>))
<explanation for causes>
(Active? ?inactive-process)
H: Flawed conditions ?inactive-process

Figure 4.17 The general form of the active? explanations.

Procedure Active?-Explanations (theory scenario behavior failure)
Generate a set of candidate processes by selecting those processes from the inactive processes in the scenario that can cause the observed change
For each process in the candidate set
do
Hypothesize that the process is active
Construct the abstract explanation based on this hypothesis and the explanation form in figure 4.17.
Return the explanations and the hypotheses

Figure 4.18 The procedure for constructing active? explanations.

Example

There are two inactive processes in the evaporation scenario of figure 4.15 – 1) heat flow from water to the vapor and 2) heat flow from water to the stone through the wall of the container. Both theses processes can explain the observed decrease in the temperature of water if they are made active in the scenario. These two processes are hypothesized to be active and the explanations based on these hypotheses constructed according to the explanation form of figure 4.17 are shown in figure 4.19.

```
(decrease (temperature water))
   (Causes (heat-flow water stone container-path) (decrease (temperature water))
      I-[(temperature water), (A (heat-flow-rate (heat-flow water stone container-path)))]
   (Active? (heat-flow water stone container-path))
      H: Flawed conditions (heat-flow water stone container-path)

(decrease (temperature water))
   (Causes (heat-flow water vapor vapor-path) (decrease (temperature water))
      I-[(temperature water), (A (heat-flow-rate (heat-flow water vapor vapor-path)))]
   (Active? (heat-flow water vapor vapor-path))
      H: Flawed conditions (heat-flow water vapor vapor-path)
```

Figure 4.19 Explanations for the observed decrease in the temperature of water based on the abstract hypothesis active?. In the first explanation a heat flow from water to the stone is hypothesized. In the second explanation a heat flow from water to the vapor is hypothesized.

4.4.1.3. New–Process? Explanations

This type of explanation is based on the abstract hypothesis *new–process?*. A new process is hypothesized to cause the observed change. According to the hypothesis, the initial domain theory is not complete in that all the processes that occur in the domain are not specified. This new process has effects that can explain the observations and conditions that are satisfied in the failure scenario. The abstract hypothesis does not specify what the conditions and effects are. These will be determined when the new process is created and added to the theory during the refinement of this abstract hypothesis. The explanation is of the general form shown in figure 4.20. The procedure for creating new processes and explanations based on the hypothesized new processes is described in figure 4.21.

```
(<change> <quantity>)
   (Causes ?new-process (<change> <quantity>))
      H: New-process ?new-process
   (Active? ?new-process)
      H: New-process ?new-process
```

Figure 4.20 The general form of the new-process? explanations.

```
Procedure New-Process-Explanations (theory scenario behavior failure)
   Hypothesize a new process that is active and is causing the observed change
   Construct an explanation based on the hypothesis and the explanation form in figure 4.20
   Return the explanation and the hypothesis
```

Figure 4.21 The procedure for constructing new-process? explanations.

Example

In the evaporation example of figure 4.15, a new process can be hypothesized to cause the observed decrease in the temperature of water. The explanation based on this abstract hypothesis and constructed according to the explanation form described in figure 4.20 is shown in figure 4.22.

```
(decrease (temperature water))
   (Causes new-process47 (decrease (temperature water)))
      H: New-process new-process47
   (Active? new-process47)
      H: New-process new-process47
```

Figure 4.22 The explanation for the observed decrease in the temperature of water based on a hypothesized new process that is active and causing the observed decrease.

4.4.2. Explanation Construction for Failed Prediction Failures

There are three types of explanations for this failure – inactive? explanations, not-causes? explanations and unexpected-observation? explanations. The procedure for constructing these explanations is shown in figure 4.23.

```
Procedure Failed-prediction-explanations (theory scenario behavior failure)
   Collect explanations from:
      (inactive?-explanations theory scenario behavior failure)
      (not-causes?-explanations theory scenario behavior failure)
      (unexpected-observation?-explanations theory scenario behavior failure)
```

Figure 4.23 The procedure for constructing explanations for the failed prediction failures.

4.4.2.1. Inactive? Explanations

This type of explanation is based on the abstract hypothesis *inactive?*. One of the processes underlying the predicted change is hypothesized to be inactive. Therefore, the predicted change is no longer true. Under this hypothesis, the conditions of the process are incorrect and have to be modified to prevent it from being predicted to be active in the scenario. The general form of the explanation is shown in figure 4.24. The procedure for creating explanations for the observation based on the hypothesis is described in figure 4.25.

```
(constant <quantity>)
   (Causes ?active-process (<change> <quantity>))
      <explanation for causes>
   (Inactive? ?active-process)
      H: Flawed conditions ?active-process
```

Figure 4.24 The general form of the inactive? explanations.

```
Procedure Inactive?-Explanations (theory scenario behavior failure)
   For each process causing the predicted change
      do
         Hypothesize the process to be inactive
         Construct the explanation based on the hypothesis and the explanation form in
         figure 4.24
   Collect all the explanations and hypotheses
```

Figure 4.25 The procedure for constructing the inactive? explanations.

Example

Consider the dissolve example described in section 3.5.2. In this example, the initial theory predicts that the amount of the salt in the container decreases in the scenario shown in figure 4.26. There is only one active process underlying the prediction that the amount of the salt in the container decreases – dissolving of the salt into the salt solution. This process is hypothesized to be inactive, thereby, defeating the prediction. The explanation for the amount of salt remaining constant based on this hypothesis constructed according to the explanation form described in figure 4.24 is shown in figure 4.27.

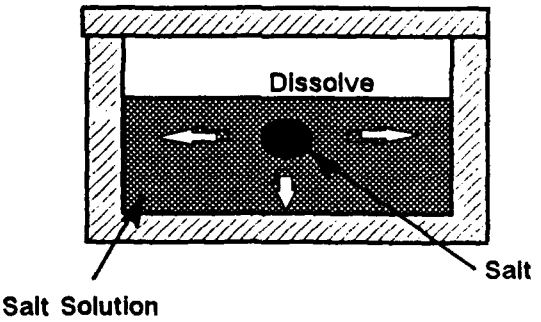


Figure 4.26 The scenario for the dissolve example.

```
(constant (amount-of salt))
   (Causes (dissolve salt-solution salt) (decrease (amount-of salt)))
      I-[(amount-of salt), (dissolve-rate (dissolve salt-solution salt))]
   (Inactive? (dissolve salt-solution salt))
      H: Flawed conditions (dissolve salt-solution salt)
```

Figure 4.27 The explanation, based on the hypothesis that the dissolve process is inactive, for why the amount of salt is constant.

4.4.2.2. Not-Causes? Explanations

This type of explanation is based on the abstract hypothesis *not-causes?*. One of the processes causing the predicted change is hypothesized to be not causing the change. According to this hypothesis, the process has incorrect effects which have to be revised so that the change is not predicted. The general form of the explanation is shown in figure 4.28. The procedure for constructing explanations based on this type of hypothesis is shown in figure 4.29.

```
(constant <quantity>)
   (Not-Causes? ?active-process (<change> <quantity>))
      H: Flawed Effects ?active-process
   (Active ?active-process)
      <Explanations for conditions>
```

Figure 4.28 The general form of the not-causes? explanations.

Procedure Not-Causes?-Explanations (theory scenario behavior failure)
  For process in active processes underlying the prediction
    do
      Hypothesize that the process does not cause the predicted change
      Construct an explanation based on the hypothesis and the explanation form in figure 4.28
  Collect all the explanations and hypotheses

Figure 4.29 The procedure for the construction of the not-causes? explanations.

Example

In the dissolve example of figure 4.26, there is only one process underlying the predicted change in the amount of salt – the dissolving of salt in the salt solution. This process is hypothesized not to cause the predicted change. The explanation based on this hypothesis constructed according to the explanation form described in figure 4.28 is shown in figure 4.30.

(constant (amount-of salt))
  (Not-Causes? (dissolve salt-solution salt) (decrease (amount-of salt)))
    H: Flawed Effects (dissolve salt-solution salt)
  (Active (dissolve salt-solution salt))
    (dissolves salt salt-solution)

Figure 4.30 The explanation, based on the hypothesis that the dissolve process does not cause a decrease in the amount of salt, for why the amount of salt is constant.

4.4.2.3. Unexpected-Observation? Explanations

This type of explanation is based on the abstract hypotheses – *unexpected-observation?* and *equals?*. The quantity is hypothesized to be influenced in the opposite direction by some other processes and the magnitude of the change is such that the two changes cancel each other exactly. According to this hypothesis, the initial theory is imperfect because it did not predict the equal and opposite change to the quantity. The general form of the explanation is shown in figure 4.31. The procedure for creating such explanations is described in figure 4.32.

```
(constant <quantity>)
   (<change> <quantity>)
      (Causes ?active-process (<change> <quantity>))
         <explanation for causes>
      (Active ?active-process)
         <Explanations for conditions>
   (<opposite-change> <quantity>)
      H: (Unexpected-observation? (<opposite-change> <quantity>))
   (equals <change> <opposite-change>)
      H: (equals? <change> <opposite-change>)
```

Figure 4.31 The general form of the unexpected-observation? explanations.

```
Procedure Unexpected-Observation?-Explanations (theory scenario behavior failure)
   Hypothesize an opposing change
   Hypothesize that the opposing change equals the predicted change
   Construct an explanation for the constant quantity based on the two hypotheses according to
   the explanation form of figure 4.31
   Return the explanation and the hypotheses
```

Figure 4.32 The procedure for constructing the unexpected-observation? explanations.

Example

In the dissolve example, the predicted change is a decrease in the amount of salt in the container. An equal increase in the amount of salt is hypothesized. The explanation based on the abstract hypotheses constructed according to the explanation form described in figure 4.31 is shown in figure 4.33.

```
(constant (amount-of salt))
   (decrease (amount-of salt))
      (Causes (dissolve salt-solution salt) (decrease amount-of salt))
         I-[(amount-of salt), (A (dissolve-rate (dissolve salt-solution salt)))]
      (Active (dissolve salt-solution salt))
         (Dissolves? salt salt-solution)
   (increase (amount-of salt))
      H: (Unexpected-observation? (increase (amount-of salt)))
   (equals (increase (amount-of salt)) (decrease (amount-of salt)))
      H: (equals? (increase (amount-of salt)) (decrease (amount-of salt)))
```

Figure 4.33 The explanation, based on the hypothesis that there is an unexpected increase in the amount of salt, for why the amount of salt is constant.

4.4.3. Explanation Construction for Inverse Behavior Failures

There are two types of explanations for this failure - causes? explanations and unexpected-observation? explanations. The procedure for constructing these explanations is

shown in figure 4.34. Note that, unlike the case of the failed prediction failure, an inactive? hypothesis -- an active process underlying the prediction with incorrect conditions - is not proposed because, then, in addition to this revision, another hypothesis that explains the observed change in the opposite direction will have to be generated.

```
Procedure Inverse-Behavior-Explanations (theory scenario behavior failure)
   Collect explanations from
      (causes?-inverse-explanations theory scenario behavior failure)
      (unexpected-observation?-inverse-explanations theory scenario behavior failure)
```

Figure 4.34 The procedure for constructing explanations for the inverse behavior failures.

4.4.3.1. Causes? Explanations

This type of explanation is based on the abstract hypothesis *causes?*. One of the processes underlying the prediction is hypothesized to cause the observed change. According to the hypothesis, the process has incorrect effects and is incorrectly predicting the change in the quantity. The effects of the process have to be revised such that the process predicts the opposite change. The general form of the explanation based on this abstract hypothesis is shown in figure 4.35. The procedure for constructing abstract explanations based on this hypothesis is shown in figure 4.36. The candidate set of process for the hypothesis is generated by selecting those active processes that cause the predicted change.

```
(<opposite-change> <quantity>)
   (Causes? ?active-process (<opposite-change> <quantity>))
      H: Flawed Effects ?active-process
   (Active ?active-process)
      <Explanations for conditions>
```

Figure 4.35 The general form of the causes? explanations.

```
Procedure Causes?-Inverse-Explanations (theory scenario behavior failure)
   Generate the candidate set of processes by selecting those processes that cause the
   observed change
   For each process in the candidate set
      do
         Hypothesize that the reverse prediction is caused by the process
         Construct an explanation for the observed change based on the hypothesis and the
         form of explanation shown in figure 4.35
   Collect all the explanations and the hypotheses
```

Figure 4.36 The procedure for constructing the causes? explanations.

Example

Consider the liquid flow example described in section 3.5.3. In this example the amount of liquid2 is predicted to decrease but is observed to increase in the scenario reproduced in figure 4.37. The

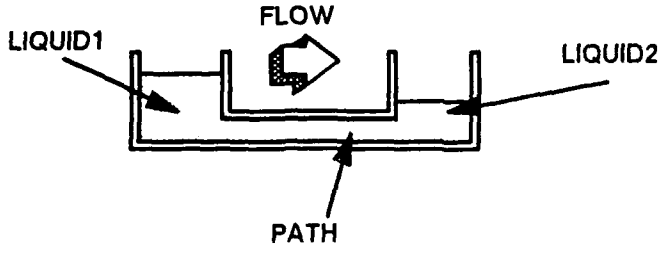


Figure 4.37 The scenario for the liquid flow example.

only process underlying the predicted change of decrease in the amount of liquid2 is the liquid flow process from liquid1 to liquid2 through the path. This process is hypothesized to cause an increase in the amount of liquid2 instead of the decrease. The explanation based on that hypothesis is shown in figure 4.38.

```
(increase (amount liquid2))
   (Causes? (liquid-flow liquid1 liquid2 path) (increase amount liquid2))
      H: Flawed Effects (liquid-flow liquid1 liquid2 path)
   (Active (liquid-flow liquid1 liquid2 path))
      (liquid-flow-aligned path)
      (greater-than (A (pressure liquid1)) (A (pressure liquid2)))
```

Figure 4.38 The explanation for the observed increase in the amount of liquid2 based on the hypothesis that the liquid flow process is causing the observed increase.

4.4.3.2. Unexpected-Observation? Explanations

This type of explanation is based on the abstract hypotheses – *unexpected-observation?* and *dominates?*. An opposite change to the quantity is hypothesized and this change is hypothesized to dominate the predicted change. According to these hypotheses, the initial theory is imperfect because it did not predict the dominating, opposite change to the quantity. The general form of the explanation based on this hypothesis is shown in figure 4.39. The procedure for constructing explanations for this type of hypothesis is described in figure 4.40.

```
(<opposite-change> <quantity>)
   (<change> <quantity>)
      (Causes ?active-process (<change> <quantity>))
         <explanation for causes>
      (Active ?active-process)
         <Explanations for conditions>
   (<opposite-change> <quantity>)
      H: (Unexpected-observation? (<opposite-change> <quantity>))
   (dominates <opposite-change> <change>)
      H: (dominates? <opposite-change> <change>)
```

Figure 4.39 The general form of the unexpected-observation? explanations.

```
Procedure Unexpected-Observation?-Inverse-Explanations (theory scenario behavior failure)
   Hypothesize an opposing change
   Hypothesize that the opposing change dominates the predicted change
   Construct an explanation for the constant quantity based on the two hypotheses and the
   explanation form in figure 4.39
   Return the explanation and the hypotheses
```

Figure 4.40 The procedure for constructing the unexpected-observation? explanations.

Example

In the flow example of figure 4.37, the amount of liquid2 is hypothesized to increase and this increase is hypothesized to dominate the decrease due to the flow process. The explanation resulting from this abstract hypothesis is shown in figure 4.41.

```
(increase (amount liquid2))
   (decrease (amount liquid2))
      (Causes (liquid-flow liquid1 liquid2 path) (decrease (amount liquid2))
         (Active (liquid-flow liquid1 liquid2 path))
            (liquid-flow-aligned path)
            (greater-than (A (pressure liquid1)) (A (pressure liquid2)))
   (increase (amount liquid2))
      H: (Unexpected-observation? (increase amount liquid2))
   (dominates (increase (amount liquid2)) (decrease (amount liquid2)))
      H: (dominates? (increase (amount liquid2)) (decrease (amount liquid2)))
```

Figure 4.41 The explanation for the observed increase in the amount of liquid2 based on the hypothesis that there is an unexpected increase and this increase dominates the predicted decrease.

4.5. Refining Hypotheses in QP Theory

The abstract hypotheses remaining after the test phase are refined. At the lowest level, refining an abstract hypothesis involves forming a new theory by making changes to the old theory that conform to the abstract hypothesis. First, the different types of theory revision operators applicable

to QP theory are described. Then, the refinement of each abstract hypothesis is described and illustrated with examples.

4.5.1. Theory Revision Operators for QP Theory

A theory revision operator specifies how a component of a new theory is revised to yield a new theory. It makes changes to the existing domain theory to produce new theories. The terminology that describes the theory revision operation is:

old-component → new-components.

The different types of theory revision operators are:

a) Adding a new component:

New components are added to the theory. The component may be a part of the process – an individual, precondition, quantity condition, relation or influence – or may be a process. For example, the operator below describes the addition of a new influence to the evaporation process:

→ **I-[(temperature ?liquid), (A (evaporation-rate ?self))].**

b) Deleting a component:

A component of the theory may be deleted. The components considered for deletion are the parts of a process – an individual, precondition, quantity condition, relation or influence. For example, the operator below describes the deletion of a precondition of the heat flow process:

(heat-aligned? ?path) →.

c) Negating a condition:

A condition (precondition or quantity condition) of a process may be negated. The negation of a quantity condition results in two quantity conditions. For example, the operator below describes the negation of a quantity condition of the heat flow process:

(greater-than (A (temperature ?source)) (A (temperature ?destination))) →
(equal-to (A (temperature ?source)) (A (temperature ?destination)))
(less-than (A (temperature ?source)) (A (temperature ?destination))).

d) Inverting an effect:

An effect of a process (a relation or influence) can be inverted by changing its type. For example, the operator below describes the inversion of a relation of the evaporation process:

(Q+ (evaporation–rate ?self) (contact–area ?liquid ?vapor)) →
(**Q–** (evaporation–rate ?self) (contact–area ?liquid ?vapor)).

e) Widening the scope of a component:

The scope of a component of the theory can be widened by making it applicable over a wider range of situations. The components considered are the parts of a process – a precondition, quantity condition, relation or an influence. There are two types of widening the scope:

1) Relacing a part by a whole: For example, if an influence affects only a part of an individual, then replacing the part by the whole individual will result in the influence affecting the other parts of the individual as well. The operator below describes how an influence of evaporation affecting only the solvent part of a solution is changed to affect the whole solution.

I–[(amount–of (solvent–of ?solution)), (A (evaporation–rate ?self))] →
I–[(amount–of **?solution**), (A (evaporation–rate ?self))].

2) Climbing an is–a hierarchy: A predicate or quantity is replaced by a more abstract quantity. For example, if an is–a relation holds between two predicates, then the more abstract predicate can be substituted for the specific abstract. The relation will then apply to other predicates that have an is–a relationship with the abstract predicate. The operator below describes how a precondition of heat flow which covers only a particular type of alignment (conducting heat) of the path can be extended to cover different types of alignment (conducting heat, electricity etc.):

(heat–aligned? ?path) → (**aligned?** ?path).

f) Narrowing the scope of a component:

This is the inverse operator for the above type of revision. It involves specializing components of the theory. The components considered are the parts of a process – a precondition, quantity condition, relation or an influence. As above, there are two types of specialization considered –

specialization based on part-whole relationships and specializations based on is-a relationships. Examples of these are the reverse of the examples illustrated above:

I-[(amount-of ?solution), (A (evaporation-rate ?self))] →
I-[(amount-of **(solvent-of ?solution)**), (A (evaporation-rate ?self))]
(aligned? ?path) → **(heat-aligned?** ?path).

4.5.2. Refining Abstract Hypotheses

If an abstract hypothesis passes the testing stage then it has to be refined to the set of hypotheses that corresponds to the abstract hypothesis. The abstract hypothesis guides both the selection of the parts of the theory to be revised and the selection of the operators to be applied. The refinement stage of each abstract hypothesis is described below:

4.5.2.1. Active?

Abstract hypothesis:	(Active? ?process)
Parts to be revised:	Failed conditions
Revision types:	Delete, Widen scope, Negation etc.
Revision Constraints:	Every failed condition must be revised

The conditions of the process that is hypothesized to be active are revised in such a manner that the process becomes active in the failure scenario. Only those conditions that failed originally have to be revised. The conditions that succeeded are not revised because they are not responsible for the process being incorrectly inactive. All of the failed conditions have to be revised so that they are satisfied in the failure scenario before the process can become active. Some of the possible revisions include deleting some of the failed conditions, negating some of the failed conditions, generalizing the range of some of the failed conditions. Revisions such as adding new conditions to the process are not suggested because they do not make the process active.

Evaporation Example

In the evaporation example of figure 4.15, two abstract hypotheses of the type *Active?* were proposed to explain the decrease in the temperature of water. As an example, consider the refinement of one of these hypotheses:

(Active? (heat-flow water stone container-path)).

The parts to be revised in the heat-flow process are those conditions that failed in the evaporation scenario of figure 4.15. The heat flow process has two conditions – a precondition requiring the path

to be heat–aligned (that is, conducting heat) and a quantity condition that requires the temperature of the source to be greater than the temperature of the destination. The only condition that failed is the heat–aligned condition since the container–path is described to be insulated against heat flow. This condition has to be revised in such a manner that the revised condition is satisfied in the evaporation scenario. Some of the revisions are:

a) Delete Condition:

The condition is deleted from the process definition allowing the process to become active in cases where this condition fails and the other conditions are satisfied. In the example, the heat aligned condition is deleted from the process definition.

(heat–aligned? ?path) →.

The heat flow process definition after the revision is made is shown in figure 4.42 and the explanation for the observed decrease in the temperature of water is shown in figure 4.43.

```
Heat–Flow (?source ?destination ?path)
   Individuals
      ?source ?destination ?path
   Preconditions
   Quantity Conditions
      (greater–than (A (temperature ?source)) (A (temperature ?destination)))
   Relations
      (Q+ (heat–flow–rate ?self) (temperature ?source))
      (Q– (heat–flow–rate ?self) (temperature ?destination))
      (Q+ (heat–flow–rate ?self) (cross–sectional–area ?path))
      (Q– (heat–flow–rate ?self) (length ?path))
   Influences
      I–[(amount–of ?source), (A (heat–flow–rate ?self))]
      I+[(amount–of ?destination), (A (heat–flow–rate ?self))]
```

Figure 4.42 The heat flow process after the precondition heat–aligned? is deleted.

```
(decrease (temperature water))
   I–[(temperature water), (A (heat–flow–rate (heat–flow water stone container–path)))]
      (Active (heat–flow water stone container–path))
         H: Delete Condition (heat–aligned? container–path)
         (greater–than (A (temperature water)) (A (temperature stone)))
```

Figure 4.43 The explanation for the observed decrease in the temperature of water based on the revised heat flow process of figure 4.42.

b) *Negate Condition:*

The condition is negated in the process definition. If the negation of the condition is true then the process will become active if the other conditions are satisfied. In the example, it is hypothesized that instead of requiring a heat-aligned path for heat-flow what is actually required is a path that is not heat-aligned (i.e. insulated).

(heat-aligned? ?path) → (:not (heat-aligned? ?path)).

The process definition based on this hypothesis is shown in figure 4.44 and the explanation based on the new revised theory is shown in figure 4.45.

```
Heat-Flow (?source ?destination ?path)
   Individuals
      ?source ?destination ?path
   Preconditions
      (:not (heat-aligned? ?path))
   Quantity Conditions
      (greater-than (A (temperature ?source)) (A (temperature ?destination)))
   Relations
      (Q+ (heat-flow-rate ?self) (temperature ?source))
      (Q- (heat-flow-rate ?self) (temperature ?destination))
      (Q+ (heat-flow-rate ?self) (cross-sectional-area ?path))
      (Q- (heat-flow-rate ?self) (length ?path))
   Influences
      I-[(amount-of ?source), (A (heat-flow-rate ?self))]
      I+[(amount-of ?destination), (A (heat-flow-rate ?self))]
```

Figure 4.44 The heat flow process after the precondition heat-aligned is negated.

```
(decrease (temperature water))
   I-[(temperature water), (A (heat-flow-rate (heat-flow water stone container-path)))]
      (Active (heat-flow water stone container-path))
         H: Negate-Condition (heat-aligned? container-path)
         (:not (heat-aligned? container-path))
         (greater-than (A (temperature water)) (A (temperature stone)))
```

Figure 4.45 The observed decrease in the temperature of water based on the revised heat flow process of figure 4.44.

c) Widening the Scope of a Condition:

The scope of the condition is widened to allow it to be satisfied in more situations including the failure scenario. Examples of widening the scope of the condition include generalizing the predicate and generalizing the arguments of the predicate. In the example, it is hypothesized that any form of

alignment will be adequate for the heat flow process. For example, heat-aligned is a special form of the aligned predicate. Other forms of alignment of paths include liquid-flow-aligned and electricity-aligned.

(heat-aligned? ?path) → (aligned? ?path)

The new revised process definition is shown in figure 4.46 and the explanation generated based on the new theory is shown in figure 4.47.

```
Heat-Flow (?source ?destination ?path)
   Individuals
      ?source ?destination ?path
   Preconditions
      (aligned? ?path)
   Quantity Conditions
      (greater-than (A (temperature ?source)) (A (temperature ?destination)))
   Relations
      (Q+ (heat-flow-rate ?self) (temperature ?source))
      (Q- (heat-flow-rate ?self) (temperature ?destination))
      (Q+ (heat-flow-rate ?self) (cross-sectional-area ?path))
      (Q- (heat-flow-rate ?self) (length ?path))
   Influences
      I-[(amount-of ?source), (A (heat-flow-rate ?self))]
      I+[(amount-of ?destination), (A (heat-flow-rate ?self))]
```

Figure 4.46 The heat flow process definition after the precondition heat-aligned has been replaced by the aligned precondition.

```
(decrease (temperature water))
   I-[(temperature water), (A (heat-flow-rate (heat-flow water stone container-path)))]
      (Active (heat-flow water stone container-path))
         H: Widen-Scope (heat-aligned? container-path)
         (aligned? container-path)
         (greater-than (A (temperature water)) (A (temperature stone)))
```

Figure 4.47 The explanation for the observed decrease in the temperature of water based on the revised heat flow process definition shown in figure 4.46.

An example of widening the scope of the argument of the condition predicate would be to substitute the whole for a part in cases in which the argument is a part. An example of that substitution would be replacing the (solute-of solution) argument by the solution.

4.5.2.2. Causes?

Abstract hypothesis: (Causes? ?process ?observation)
Parts to be revised: Effects of ?process

Revision types: Add new effects, Narrow/Widen Scope
Revision Constraints: At least one effect must be revised

The effects of the process that is hypothesized to cause the observed change are revised so that the process causes the observed change in the failure scenario. Some of the possible revisions include adding new influences or relations and modifying existing influences or relations by specializing or generalizing or inverting.

Evaporation Example

In the evaporation example, a hypothesis of the type Causes? is proposed to explain the unexpected decrease in the temperature of water in the scenario shown in figure 4.15. Consider the refinement of that abstract hypothesis:

(Causes? (evaporation water vapor) (decrease (temperature water)))

The effects of the evaporation process are to be revised. Some of the revisions include:

a) New Influence:

A new influence is added to the process. The new influence may explain the unexpected observation in two ways: 1) directly, by causing the change in the influence itself or 2) indirectly, by explaining the observed change in conjunction with other influences and relations that are true of the scenario. For example, if the observed change is an increase in the quantity Q. Then the direct method would involve hypothesizing a new influence that affects the quantity Q. The indirect method would involve identifying a relation of the type

(Qprop Q Q1)

which is valid in the scenario and hypothesizing a new influence on the quantity Q1.

An example of a direct influence is:

→ I-[(temperature ?liquid), (A (evaporation-rate ?self))].

The new process definition for evaporation incorporating this hypothesized change is shown in figure 4.48 and the explanation based on this revised theory is shown in figure 4.49.

```
Evaporation (?liquid ?vapor)
   Individuals
      ?liquid ?vapor
   Preconditions
      (open? (container ?liquid))
   Quantity Conditions
   Relations
      (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))
   Influences
      I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
      I+[(amount-of ?vapor), (A (evaporation-rate ?self))]
      I-[(temperature ?liquid), (A (evaporation-rate ?self))]
```

Figure 4.48 The process definition for evaporation after the new influence has been added.

```
(decrease (temperature water))
   I-[(temperature water), (A (evaporation-rate (evaporation water vapor)))]
      (Active (evaporation water vapor))
         (open? (container water))
   H: New-Influence? I-[(temperature water),(A (evaporation-rate (evaporation water
   vapor)))]
```

Figure 4.49 The explanation for the observed decrease in the temperature of water based on the revised definition for evaporation shown in figure 4.48.

b) Adding a New Relation:

A new relation can be added to explain the observed change. The quantities that are changing due to the process can be determined and used to construct hypothesized relations linking the observed change to each of the quantities changing. For example, if a quantity Q1 is known to be changing in the scenario and is caused by the process and Q is the observed change then the following relation can be hypothesized:

(Qprop Q Q1)

where the type of the proportionality is determined by the manner in which Q and Q1 are changing (for example, if both are increasing then it is Q+).

In the evaporation temperature example, the quantities that are changing due to the evaporation process are:

(amount-of water) → Decrease
(amount-of vapor) → Increase.

Two new relations can be proposed:

(Q+ (temperature ?liquid) (amount-of ?liquid))
(Q- (temperature ?liquid) (amount-of ?vapor)).

Consider the first revision proposed. The new revised theory for evaporation is shown in figure 4.50 and the explanation for the observed change based on this revised theory is shown in figure 4.51.

Evaporation (?liquid ?vapor)
  Individuals
    ?liquid ?vapor
  Preconditions
    (open? (container ?liquid))
  Quantity Conditions
  Relations
    (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))
    (Q+ (temperature ?liquid) (amount-of ?liquid))
  Influences
    I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
    I+[(amount-of ?vapor), (A (evaporation-rate ?self))]

Figure 4.50 The process definition for evaporation after a new relation has been added.

(decrease (temperature water))
  (Q+ (temperature water) (amount-of water))
      (Active (evaporation water vapor))
        (open? (container water))
  (decrease (amount-of water))
    I-[(amount-of water), (A (evaporation-rate (evaporation water vapor)))]
      (Active (evaporation water vapor))
        (open? (container water))
  H: New-Relation? (Q+ (temperature water) (amount-of water))

Figure 4.51 The explanation for the observed decrease in the temperature of water based on the revised evaporation definition shown in figure 4.50.

4.5.2.3. Inactive?

Abstract Hypothesis:	(Inactive? ?process)
Parts to be revised:	Conditions of ?process
Types of revision:	Adding new conditions, Narrowing Scope, Negation
Constraints:	At least one condition must be revised so that it is unsatisfied in the failure scenario.

The conditions of the process that is hypothesized to be inactive are revised so that the process becomes inactive in the given scenario. Some of the possible revisions include adding new preconditions or quantity conditions that are not satisfied in the given scenario and specializing or

```
(<change> <quantity>)
   (Causes? ?active-process (<change> <quantity>))
      H: Flawed effects ?active-process
   (Active ?active-process)
      <Explanation for conditions>
```

Figure 4.13 The general form of the causes? explanations.

```
Procedure Causes?-Explanations (theory scenario behavior failure)
   Generate a set of candidate processes from the set of active processes
   For each process in the candidate set
      do
         Hypothesize that the process causes the observed change
         Construct the abstract explanation based on this hypothesis and the form of the
         explanation in figure 4.13
   Return the explanations and the hypotheses
```

Figure 4.14 The procedure for constructing the causes? explanations.

Example

Consider the evaporation example described in section 3.5.1. In this example an unexpected decrease in the temperature of water is observed in the scenario (reproduced in figure 4.15). The

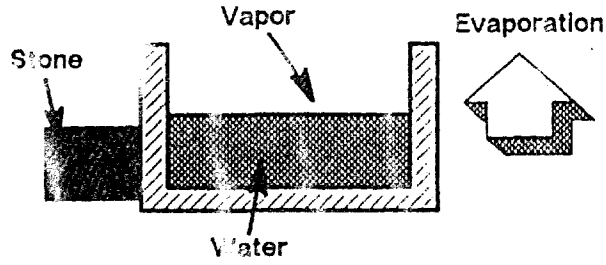


Figure 4.15 The scenario for the evaporation example.

only active process is the evaporation of water. This process is hypothesized to cause the decrease in the temperature of water. The explanation for this change based on this hypothesis and constructed according to the explanation form of figure 4.13 is shown in figure 4.16.

```
Dissolve (?solution ?solid)
   Individuals
      ?solution ?solid
   Preconditions
      (dissolves? ?solid ?solution)
      (new-precondition2457 ?solution ?solid)
   Quantity Conditions
   Relations
      (Q+ (dissolve-rate ?self) (contact-area ?solution ?solid))
   Influences
      I-[(amount-of ?solid), (A (dissolve-rate ?self))]
      I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]
```

Figure 4.52 The process definition for dissolve obtained by adding a new precondition.

```
(constant (amount-of salt))
   (Causes (dissolve salt-solution salt) (decrease (amount-of salt)))
      I-[(amount-of salt), (A (dissolve-rate (dissolve salt-solution salt)))]
   (Inactive? (dissolve salt-solution salt))
      (not (new-precondition4857 salt-solution salt))
      H: New precondition (new-precondition4857 salt-solution salt)
```

Figure 4.53 The explanation for why the amount of salt is not changing based on the revised dissolve definition of figure 4.52.

b) *New Quantity Condition*

The quantities that are changing in the given scenario are:

(amount-of salt) → Decrease
(amount-of (solute-of salt-solution)) → Increase
(concentration salt-solution) → Increase.

Three quantity conditions based on quantities reaching a limit point can be proposed. They are:

(amount-of salt) > (Minimum-amount-of-point salt)
(amount-of (solute-of salt-solution))
< (maximum-amount-of-point (solute-of salt-solution))
(concentration salt-solution) < (maximum-concentration-point salt-solution).

There are two quantities of similar type that are changing and the quantity condition resulting from those quantities are:

(amount-of salt) > (amount-of (solute-of salt-solution)).

The theory resulting from the first saturation limit point is shown in figure 4.54 and the explanation from this theory for the failed prediction is shown in figure 4.55.

```
Dissolve (?solution ?solid)
   Individuals
      ?solution ?solid
   Preconditions
      (dissolves? ?solid ?solution)
   Quantity Conditions
      (greater-than (A (amount-of ?solid)) (A (minimum-amount-of-point ?solid)))
   Relations
      (Q+ (dissolve-rate ?self) (contact-area ?solution ?solid))
   Influences
      I-[(amount-of ?solid), (A (dissolve-rate ?self))]
      I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]
```

Figure 4.54 The process definition for dissolve after a new quantity condition is added.

```
(constant (amount-of salt))
   (Causes (dissolve salt-solution salt) (decrease (amount-of salt)))
      I-[(amount-of salt), (A (dissolve-rate (dissolve salt-solution salt)))]
   (Inactive? (dissolve salt-solution salt))
      (equal-to (A (amount-of salt)) (A (minimum-amount-of-point salt)))
      H: New quantity condition (greater-than (A (amount-of salt))
                                              (A (minimum-amount-of-point salt)))
```

Figure 4.55 The explanation for why the amount of salt is not changing based on the revised definition for dissolve shown in figure 4.54.

4.5.2.4. Not-causes?

Abstract Hypothesis:	(Not-Causes? ?process ?observation)
Parts to be revised:	Effects underlying the predicted change
Types of revisions:	Delete, Specialize
Constraints:	The revised effects must not lead to prediction

The effects of the process that is hypothesized not to cause the predicted change are revised so that it does not predict the change. Only those effects that were involved in the prediction are examined. Some of the revisions that are possible are deleting or specializing the offending effects.

Dissolve Example:

In the dissolve example of figure 4.26, a hypothesis that the dissolve process does not cause the predicted change – decrease in the amount of the salt – is proposed. This hypothesis can be refined by deleting or specializing the effects of dissolve in such a manner as to prevent the prediction.

Only one effect of the dissolve process is involved in the explanation for the predicted change – the influence concerning the amount of salt:

I-[(amount-of ?solid), (A (dissolve-rate ?self))].

For example, the influence can be deleted. The resulting revised theory for dissolve is shown in figure 4.56. The explanation for the prediction is shown in figure 4.57.

I-[(amount-of ?solid), (A (dissolve-rate ?self))] →.

Dissolve (?solution ?solid)
  Individuals
    ?solution ?solid
  Preconditions
    (dissolves? ?solid ?solution)
  Quantity Conditions
  Relations
    (Q+ (dissolve-rate ?self) (contact-area ?solution ?solid))
  Influences
    I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]

Figure 4.56 The process definition for dissolve after the influence has been deleted.

(constant (amount-of salt))
  (Not-Causes (dissolve salt-solution salt) (decrease (amount-of salt)))
    H: Delete-Influence I-[(amount-of salt), (A (dissolve-rate (dissolve salt-solution salt)))]
  (Active (dissolve salt-solution salt))
    (dissolves salt salt-solution)

Figure 4.57 The explanation for why the amount of salt is unchanged based on the revised process definition for dissolve shown in figure 4.56.

4.5.2.5. Equals?

This abstract hypothesis is not refined.

4.5.2.6. Dominates?

This abstract hypothesis is not refined.

4.5.2.7. Unexpected-observation?

Abstract Hypothesis: (Unexpected-observation? ?change)

This abstract hypothesis is refined to the abstract hypotheses for the unexpected observation failure – causes?, active? and new-process?.

4.5.2.8. New-process?

Abstract Hypothesis:	(New-Process? ?new-process)
Parts to be revised:	?new-process
Types of revision:	New precondition and effects
Constraints:	New process must have effects that cause the predicted change and conditions such that it is active in the failure scenario.

The new process that is hypothesized to be active and causing the observed phenomenon is constructed. In addition, all the observed changes and additional information obtained during the testing of the abstract hypotheses are incorporated into the new process definition. The new process consists of five pieces of information:

1) Individuals: The individuals that participate in the new process are those objects that are used by the other parameters of the process.

2) Preconditions: The new process has a new precondition that links the individuals participating in the process and defines the relationship that must hold (and is assumed to hold in the given scenario) for the process to become active.

3) Quantity conditions: There are no quantity conditions for the new process. Quantity conditions are added to the new process when subsequent failures require this theory revision to the new process.

4) Relations: Those unexplained changes that are differential in nature are converted to relations involving the rate of the process. Changes that involve quantities that are already a part of active relations in the scenario are also formed as relations.

5) Influences: The remaining changes are incorporated into the direct effects of the process.

The process defines the precondition that is created along with the process definition. The precondition holds for those scenarios in which the process is found to be active.

Evaporation Example

In the evaporation example, a new process can be hypothesized to explain the observed decrease in the temperature of water. The unexplained changes obtained during the testing phase are:

(temperature water) → decreases
(temperature water) decreases faster when the contact-area between water and vapor is increased.

The new process is shown in figure 4.58.

```
New-process (?liquid ?vapor)
   Individuals
      ?liquid ?vapor
   Preconditions
      (new-precondition8888 ?liquid ?vapor)
   Quantity Conditions
   Relations
      (Q+ (new-process-rate ?self) (contact-area ?liquid ?vapor))
   Influences
      I-[(temperature ?liquid), (A (new-process-rate ?self))]
```

Figure 4.58 The new process definition.

The individuals of the new process are ?liquid and ?vapor since these two objects are used by the process in the effects fields. A new precondition linking these objects is created - *new-precondition8888*. The differential change observed appears as a qualitative relation involving the rate of the process. The observed change in the temperature of water appears as an influence of the process. The explanation for the observed change using this new theory is shown in figure 4.59.

```
(decrease (temperature water))
   I-[(temperature water), (A (new-process-rate (new-process water vapor)))]
      (Active (new-process water vapor))
         (new-precondition8888 water vapor)
   H: New-Process? (new-process water vapor)
```

Figure 4.59 The explanation for the observed decrease in the temperature of water based on the new process definition of figure 4.58.

4.6. Discussion

Abstraction spaces have been used in hierarchical planning by systems such as NOAH [Sacerdoti77], ABSTRIPS [Sacerdoti74] and MOLGEN [Stefik81]. The abstraction constraint described in this chapter corresponds most to the abstraction in NOAH. In this system, the operators were abstracted and planning was initially performed with generalized operators. These operators were later refined to the problem solving operators in the problem space. While the abstraction spaces were used in NOAH to postpone commitment to a sequence of operators, in

explanation–based theory revision, abstraction is used to structure the hypothesis space in order to restrict the number of hypotheses that have to be tested.

Falkenhainer [Falkenhainer88a] describes a method called *difference–based reasoning* which uses the difference between the failure scenario and a similar scenario in which the failure did not occur to form hypotheses. Difference–based reasoning requires the system to retain a collection of past scenarios. The hypothesis generation described in this chapter does not require such a memory. However, if a history of previous experiences is collected then difference–based reasoning can be advantageously applied to further constrain the hypotheses generated.

To summarize, this chapter has described how a hypothesis generator for proposing revisions to a theory is constrained by the scenario in which the failure occurred, the need to construct an explanation for the observation in the scenario and the abstraction of hypotheses. An implementation of a hypothesis generator which proposes revisions for domain theories represented in QP theory was described. The abstract hypotheses used by the hypothesis generator and the types of explanations based on these abstract hypotheses for each type of failure was described. The refined hypotheses corresponding to each abstract hypothesis and the construction of revised theories using the refined hypotheses was also discussed.

CHAPTER 5

EXPERIMENTATION–BASED HYPOTHESIS REFUTATION

5.1. Introduction

Theory revision is invoked when a theory fails to explain the observations made in a given scenario. In general, the original theory can be revised in many different ways to eliminate the failure. Chapter 4 describes how such theories are formed based on hypothesized revisions to the original theory. Using all of these theories to predict the behavior of future scenarios is not tenable due to limitations in computational resources. Consequently, one theory (or a small number of theories) must be selected. The selection cannot be made arbitrarily or based solely on syntactic criteria. Many of the proposed theories will be incorrect. Though they can correctly explain the observations made in the given scenario, they may not be able to explain future observations. The selection of an incorrect theory can have disastrous consequences if the predictions are used to govern critical decisions such as shutting down a nuclear reactor or a steam plant. Therefore, it is important to expend additional resources to identify incorrect theories and eliminate them.

One human method to combat multiple hypotheses is to actively interact with the environment. The interactions yield new data from the world which may eliminate some of the hypotheses. Even when there are relatively few hypotheses, active pruning may be desirable. Humans often perform cheap tests to head off undesirable possibilities. They heft a snowball several times before throwing it or test the swimming pool temperature with a toe before diving in.

This chapter describes a method called *experimentation–based hypothesis refutation* for testing hypotheses. Experimentation–based hypothesis refutation involves the purposeful interaction with the world in such a way that observable behavior will dictate which of a number of incompatible hypotheses corresponds to reality. This process can be looked upon as conducting experiments in

the world. The outcome of these experiments provide additional data which, provided the experiments are suitably chosen, will be inconsistent with a number of hypotheses. These hypotheses can then be eliminated from further consideration.

Experimentation–based hypothesis refutation is a method that actively seeks additional information to refute theories. It involves manipulating the world and measuring quantities whose values are not known. In contrast, a passive method relies on fortuitous observations to provide the information that is required to eliminate the incorrect theories. Passive methods do not have the overhead cost of designing and conducting experiments. However, active methods have the potential to avert catastrophes by performing simple experiments to identify and eliminate theories that make incorrect predictions. In the case of passive methods, the first clue that the theory is incorrect can be the catastrophe itself.

Experimentation–based hypothesis refutation is a focused quest for information. The predictions made by each hypothesis are analyzed and only those experiments that can yield information useful in refuting one or more of the hypotheses are designed. In contrast, other methods, that also change the world or make measurements, can accidentally stumble on the information that will refute a hypothesis.

The experiment design process must have several important features. First, it must be efficacious; if there is a way to tease apart different hypotheses the design system should find it. Second, it must be tolerant of unavailable data. Third, it must be efficient. Each experiment should evenly divide the hypotheses so that significant information is acquired regardless of the experiment's outcome. Fourth, it must be practical. Lighting a match is not a reasonable way to tell whether a nearby barrel contains water or gasoline.

As an example of experimentation–based hypothesis refutation, consider the scenario shown in figure 5.1a. In this scenario, some salt and a solution of salt in water are placed in an open container. A stone, which is at a much lower temperature than that of the solution, touches a wall of the container. The wall of the container is insulated against heat flow. The initial theory of the system predicts that the solution evaporates and that the salt dissolves in the solution. However, in the initial theory, the evaporation process is believed not to affect the temperature of the dissolving liquid

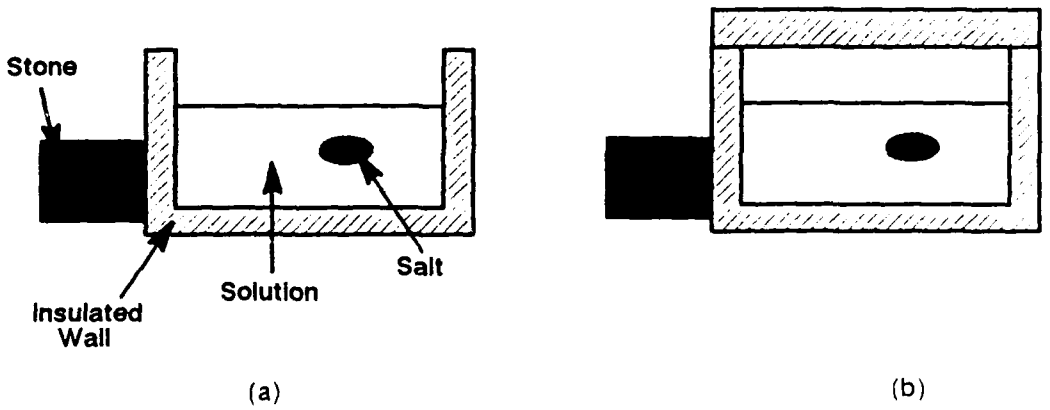


Figure 5.1 (a) A scenario in which a solution of salt in water and some salt are placed in an open container. A stone which is at a lower temperature than that of the solution is in contact with the wall of the container. The wall of the container is insulated against heat flow. (b) The scenario obtained by closing the container in the first scenario.

and, consequently, the initial theory cannot explain an observed decrease in the temperature of the solution. Some of the revisions to the initial theory that can explain this observation are:

1) Despite the insulated wall, there is a heat flow from the solution to the stone resulting in a decrease in the temperature of solution.

2) The evaporation of the solution causes a decrease in the temperature of the solution.

3) The dissolving of the salt causes a decrease in the temperature of the solution.

Experimentation-based hypothesis refutation designs experiments to test the hypotheses by analyzing the predictions made by each hypothesis. The first hypothesis, which involves a flow of heat to the stone, predicts that the temperature of the stone increases in the scenario shown in figure 5.1a. The other two hypotheses predict that the temperature of stone remains constant. Thus, an experiment to measure the temperature of the stone determines which of the hypotheses is not correct. Also, an experiment involving the construction of a new scenario in which the container is closed (figure 5.1b) can be designed. In this scenario, the second hypothesis predicts that the temperature of the solution remains constant because evaporation is no longer active.

However, the third hypothesis predicts that the temperature of the solution decreases as in the original scenario since closing the container does not affect the dissolving of the salt in the solution. Hence, measuring the temperature of the solution in the new scenario determines which hypothesis is not correct.

Notice that experimentation–based hypothesis refutation exploits the fact that each hypothesis makes definite predictions about the behavior of the quantities to design experiments. It is not necessary for the system to wait passively hoping that the change in the temperature of the stone will be measured. Nor is it necessary for the system to hope that, while manipulating objects in the world, it will accidentally stumble on the scenario shown in figure 5.1b. The observation may never be made or the system may never construct the scenario since there are numerous other quantities to be observed and numerous other scenarios that can be constructed.

The next section describes experimentation–based hypothesis refutation. The third section describes an implementation of this method. The fourth section analyzes some features of the method. Finally, the fifth section discusses related work and summarizes the chapter.

5.2. Experimentation–based Hypothesis Refutation

Experimentation–based hypothesis refutation is an active, focused method for testing hypotheses It consists of three steps:

[a] Obtaining Predictions

This step involves obtaining predictions made by each hypothesized theory. The predictions specify values for quantities whose values have not yet been observed or determined. The *legal values* of a quantity are the values that are theoretically feasible for a quantity. The *permissible values* of a quantity are the values (from the legal values) that are consistent with the information already known about the quantity. A *prediction* of a theory is a statement that specifies the values for a quantity (from the permissible values) that are supported by the theory.

Prediction[Theory, Quantity]: Value(Quantity) = $\{V_1, V_2, \ldots Vk\}$ where each Vi is a permissible value.

A prediction is *definite* if it predicts a single value, *ambiguous* if it predicts more than one value, and *agnostic* if it predicts all the permissible values.

[b] Designing Experiments

Experiments are designed based on an analysis of the obtained predictions and according to the classification described in section 5.2.1. An experiment determines the *experimental values* for a measurable quantity.

Experiment: Value(Quantity) = $\{V_1, V_2, \ldots Vm\}$ where each Vi is a permissible value.

An experiment is *definite* if the quantity has only one experimental value, *ambiguous* if it has more than one, and *inconclusive* if the experimental values are identical to the permissible values prior to the experiment.

[c] Refuting Hypotheses

Hypotheses whose predictions are not compatible with the experimental results are eliminated. A prediction is incorrect if all the predicted values are incompatible with each of the experimentally determined values. A hypothesis that makes an incorrect prediction is incorrect and is rejected.

$\forall$ $v_p \in V_p$, $\forall$ $v_e \in V_e$ Prediction[Hypothesis, Quantity]: Value(Quantity) = v_p
& Experiment: Value(Quantity) = v_e & Incompatible(v_p, v_e)
$\supset$ Incorrect(Prediction[Hypothesis, Quantity]). (V_p is the set of predicted values and V_e is the set of experimentally determined values.)

$\forall$ h $\in$ H Incorrect(Prediction[h, Quantity]) $\supset$ Incorrect(h). (H is the set of hypotheses).

In general, experimentation to test hypotheses is performed based on auxiliary assumptions The auxiliary assumptions that form the basis for experimentally testing the theory revision hypotheses are (also please refer to section 1.2): the scenarios are correctly described and the inferencing mechanism is correct. Both these assumptions are made during the elimination of hypotheses.

There are two circumstances in which experimentation-based hypothesis refutation fails:

1) Indistinguishable hypotheses: Hypotheses are indistinguishable from one another if they always make identical predictions. Such hypotheses cannot be distinguished from one

another based on external behavior. An example of indistinguishable hypotheses is alternative theories that are syntactic variants of each other.

2) Compatible hypotheses: Hypotheses are compatible if they always make predictions that are consistent with the known information. In this case, the hypotheses can make different predictions but experimentation fails because a quantity cannot be measured or because the measurements are not sufficiently accurate to determine which of the incompatible hypotheses are incorrect.

5.2.1. Strategies for Experiment Design

There are three strategies for designing experiments to refute hypotheses:

[a] Elaboration

In a given domain, there are usually some quantities that are easily measurable. In *elaboration*, the quantity to be measured is selected according to the ease with which it can be measured. Hypotheses that predict values for the quantity which are not compatible with the measured value can be immediately refuted. This strategy does not guarantee that the designed experiment will refute a hypothesis since all the hypotheses may predict the observed value or may not make any predictions regarding the value of the measured quantity.

[b] Discrimination

In *discrimination* the differences in the predictions made by the hypotheses are exploited to design experiments. If the values predicted for a quantity by a set of hypotheses are incompatible with the values predicted by another set of hypotheses then the quantity is called a *discriminant*. A discrimination experiment measures the value of a discriminant. A discrimination experiment is guaranteed to refute hypotheses because the experimental results can be compatible with only one set of predicted values. Hypotheses that predict values from the other sets are incorrect and can be eliminated based on the discrimination experiment. Discrimination is more effective than elaboration, but can be much more expensive because it has the additional overhead of comparing the predictions of the hypotheses. Similar forms of discrimination techniques have been previously utilized in diagnosis (for example, INTERNIST[Miller82]) to discriminate among competing hypotheses in order to locate a set of faults in an artifact or a set of diseases in a patient.

Figure 5.2 shows the space of values for a quantity. The quantity is a discriminant because the

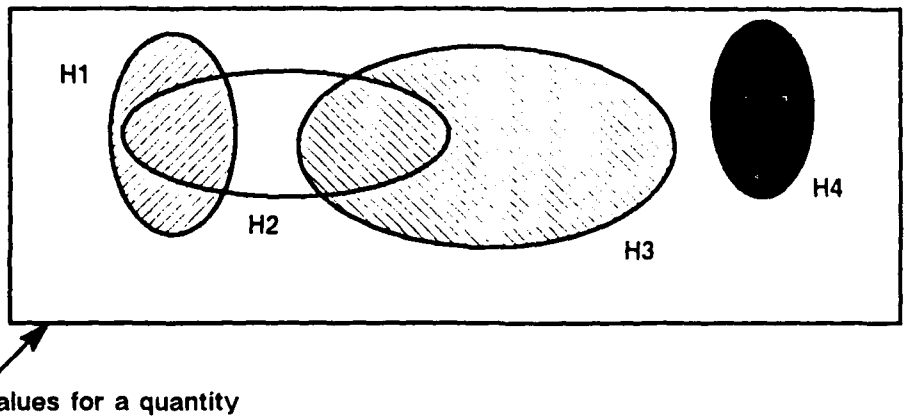


Figure 5.2 The space of values for a quantity and the values predicted by each hypothesis.

values predicted for the quantity by hypothesis H4 are incompatible with the values predicted by the other hypotheses. If the value of the quantity is determined through experimental measurement to be among the values predicted by H4 then the other three hypotheses, H1, H2 and H3, can be eliminated.

[c] Transformation

It may not be possible to identify the correct hypothesis even after all the measurable quantities in the scenario have been measured. In this case, all the remaining hypotheses make predictions that are consistent with the information that can be experimentally obtained from the scenario. However, this does not mean that more incorrect hypotheses cannot be refuted. *Transformation* of scenarios is a powerful technique of experiment design which involves creating new scenarios in which the predictions of the hypotheses can be tested. It involves modifying the original scenario in a well-specified manner, such as, changing the values of the properties of the components of the scenario (for example, changing the concentration or amount of a solution), replacing components (for example, replacing a container with a heat-insulated container) or re-organizing the manner in which the components are put together (for example, separating two containers originally connected by a pipe). The techniques of elaboration and discrimination can then be used on the new scenario to refute hypotheses. There are three important consequences of creating a new scenario:

(1) **Divergence of values:**

Hypotheses that previously predicted identical or compatible values for a quantity may predict different or incompatible values in the new scenario. Figure 5.3 illustrates the

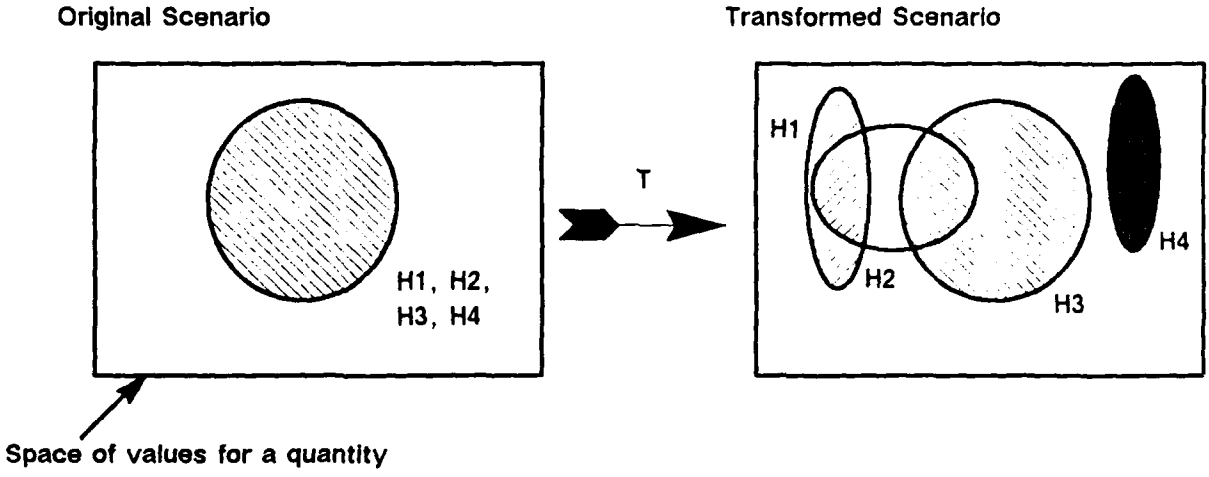


Figure 5.3 The space of values for a quantity in the original scenario and the transformed scenario. In the transformed scenario, the values for the quantity predicted by each hypothesis are not identical as in the original scenario.

divergence of the values supported by each hypothesis in the transformed scenario.

(2) **Emergent observations:**

Quantities that could not be previously measured become measurable in the new scenario. Consider, for example, tracing an unknown path of flowing water. If a dye is added to the original water at the source, then the color changes in the water may enable the path to be traced.

(3) **Differential discrimination:**

Differential discrimination involves the measurement of a quantity that satisfies the following two criteria: 1) There are a number of hypotheses that predict the same value for the quantity in the original and the transformed scenarios. 2) The predictions for the manner in which the value is reached in the transformed scenario as compared with the manner in which it is reached in the original scenario are different and incompatible for some of the hypotheses. The observations may be reached much faster or more of the observed

behavior may occur in the same time span as compared to the other scenario. Thus differential discrimination involves the discrimination of the magnitude of the change in the quantity in each scenario.

5.3. Experimentation–based Hypothesis Refutation in COAST

The first subsection describes an experiment engine – an implementation of experimentation–based hypothesis refutation – used by COAST. The remainder of the section describes how the theory revision hypotheses developed in chapter 4 for theories represented in QP theory are tested using the experiment engine. The second subsection describes how predictions for the different types of hypotheses are obtained. The third subsection describes the procedure for each of the three strategies described in the earlier section and provides a description of the different types of domain knowledge required by each strategy.

5.3.1. An Experiment Engine

An experiment engine has been developed based on the model of experimentation described above. Figure 5.4 shows the architecture of the experiment engine. Figure 5.5 describes the top

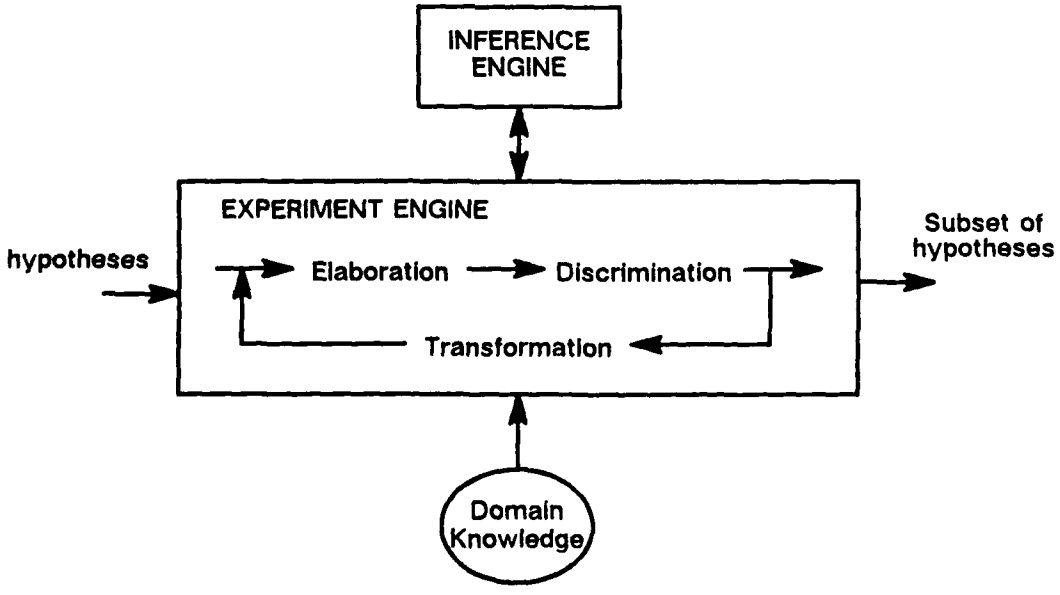


Figure 5.4 The architecture of the experiment engine.

level procedure of the experiment engine.

```
Procedure Experimentation (hypotheses scenario)
   original-scenario = scenario
   Do
      Obtain predictions supported by each hypothesis for the scenario
      Elaboration Experiments
      Discrimination Experiments
      Obtain differential predictions between the scenario and the original scenario supported by
      each hypothesis
        ;;; COAST performs differential experiments only for the transformed scenario with
        ;;; respect to the original scenario.
      Differential Elaboration Experiments
      Differential Discrimination Experiments
      If the number of valid hypotheses remaining is less than or equal to 1
       or the resource limit is exceeded
        ;;; The resource limit is a user-supplied parameter that limits the number of scenarios
        ;;; investigated by the system.
        then exit the do loop with the valid hypotheses
        else scenario = Transformation (scenario valid-hypotheses)
           If no new scenario available from transformation
                ;;; Space of scenarios available through transformation exhausted.
              then exit the do loop with the valid hypotheses
```

Figure 5.5 The procedure for designing experiments.

The inputs to the experiment engine are a set of hypotheses, an inference engine that accepts a hypothesis and a scenario and returns a set of predictions supported by the hypothesis for the given scenario and domain knowledge. In the case of theory revision, each hypothesis specifies a set of revisions to the original theory and the inference engine computes the behavior predicted by the revised theory for the given scenario. Two types of domain knowledge are required: [a] A set of predicates that describe the quantities of the domain that can be measured, the quantities of the domain that are easily measurable, the values for each quantity that are incompatible, and the parameters of a scenario that can be manipulated for transformation. [b] A set of scenario transformation operators. A scenario transformation operator constructs a new scenario from a given scenario. It can change the quantities of some of the components of the scenario eg. the concentration of a solution; the components of the scenario eg. replacing a solution by a different solution; or the manner in which the components are organized eg. isolating two containers which were previously connected by a pipe. These operators endow the experiment designer with the ability to construct new scenarios. The experiment engine designs experiments to test the hypotheses and returns those hypotheses that are consistent with the information obtained from the experiments.

The experiment engine uses elaboration, discrimination and transformation to design experiments. Transformation of scenarios is viewed as a planning problem. The initial state is the given scenario and the goal state is a transformed scenario in which there is a discriminable new observation, divergent values or discriminable first order or second order behavior. The plan is a sequence of transformations resulting in a scenario satisfying the goal criterion. Both a weak method planning strategy (Breadth-First Search) and a knowledge-intensive strategy (based on Qualitative Process theory [Forbus84b]) have been implemented.

5.3.2. Obtaining Predictions for Theory Revision Hypotheses

Predictions are obtained from the inference engine which computes the behavior of a given scenario using a given theory. Note that using all the predictions obtained from the revised theory may not be practical. Instead only those predictions that depend on the revised components of the theory need be considered since they are important for distinguishing one hypothesized theory from another. Three types of computations are involved in obtaining predictions for experimentation:

[a] Predictions for Abstract Hypotheses

Abstract hypotheses are based on the processes that were instantiated for the given scenario. The inference engine computes the behavior for the given scenario by using the original theory along with the conditions imposed by the abstract hypothesis. The imposed conditions depend on the type of the abstract hypothesis. For example, if the abstract hypothesis is of the *active?* type then the process is assumed to be active and its effects are added during the computation of the behavior. If the hypothesis is of the *inactive?* type then the process is assumed to be inactive and its effects are not added during the computation of the behavior.

Consider the abstract hypothesis:

(active? (heat-flow water stone container-path))

described in section 4.4.1.2 that is proposed for the evaporation example of figure 5.6. All the effects of the heat flow process are added during the computation of the behavior for the scenario shown in figure 5.6. The effects of the heat flow process include 1) the direct influences on the temperatures of the stone and the water and 2) the dependence of the rate of the heat flow process on the temperature of the stone, the temperature of the water, the length of the path

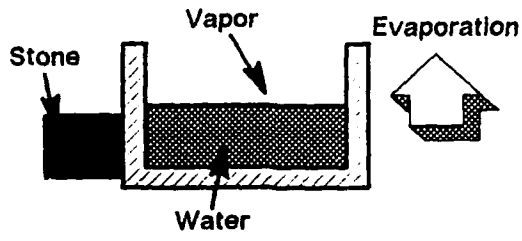


Figure 5.6 A scenario in which water is placed in an open container in contact with its vapor. A stone which is at a lower temperature than the water is in contact with the wall of the container. The wall is insulated against heat flow.

through the container and cross-sectional area of the path through the container. Some of the resulting predictions for the scenario based on the hypothesis are:

(increase (temperature stone))
(decrease (temperature water)).

[b] Predictions for Concrete Hypotheses

In the case of the refined hypotheses that propose concrete revisions to the original theory, the theory corresponding to the revisions can be constructed. The inference engine computes the behavior of the scenario based on the revised theory. As an example, consider the hypothesis:

New Relation for Evaporation: (Q+ (temperature ?liquid) (amount-of ?liquid))

for the evaporation example. The predictions for the scenario shown in figure 5.6 obtained using the revised theory include:

(decrease (amount-of water))
(decrease (temperature water)).

[c] Predictions for Transformed Scenarios

There are two types of transformations: 1) changing the status of facts (for example, constructing a new scenario in which a precondition of a process is not satisfied), and 2) making the value of a quantity greater than or less than what it is in the original scenario (for example, constructing a new scenario in which the length of a path is increased). In the case of transformations involving changes to the status of the facts in the original scenario, the transformed scenario is constructed

from the original scenario by adding the changes to the facts of the original scenario. The behavior of the transformed scenario is computed by the inference engine as above for abstract or refined hypotheses.

As an example, consider the transformed scenario shown in figure 5.7 obtained from the the

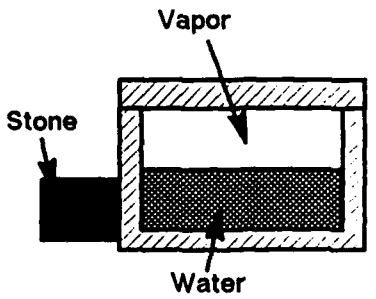


Figure 5.7 A scenario obtained by transforming the scenario shown in figure 5.6. The container is closed to prevent evaporation of the water.

scenario shown in figure 5.6 by closing the container. The transformation involves forming a new scenario that is identical to the original scenario except that the fact which specified that the container is open is retracted and the fact that the container is not open is included. The inference engine uses the new description of the layout of the scenario to compute predictions for each hypothesis. In the transformed scenario, since the container is not open evaporation of water is no longer active. The predictions supported by the abstract hypothesis:

(Causes? (evaporation water vapor) (decrease (temperature water)))

for the scenario shown in figure 5.7 include:

(constant (temperature water))
(constant (amount-of water)).

The second type of transformations involves making the value of a quantity greater than or less than the value of the quantity in the original scenario. This type of transformation is restricted to quantities that affect the rate of a process (through qualitative proportionalities) and which are not directly influenced. The predictions for the differences in the behavior of the original scenario and the transformed scenario are computed by reasoning using the qualitative proportionalities about the differences in behavior of the directly influenced quantities due to the differences in the rates of

the processes affected by the transformation. Note that, in the general case in which quantities that are directly influenced are also transformed, computing the behavior of the transformed scenarios requires *differential qualitative analysis* [Forbus84b] or *comparative analysis* [Weld87]. This type of reasoning is currently beyond the capabilities of the inference engine used by COAST.

As an example, consider the original and the transformed scenarios shown in figure 5.8. The

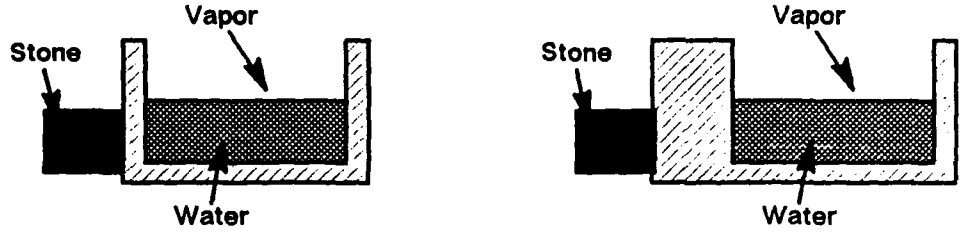


Figure 5.8 The second scenario is obtained by transformation of the first scenario. The path through the container connecting the water and the stone is increased.

transformed scenario is obtained by making the length of the path through the container greater than the length of the path in the original scenario. Consider the abstract hypothesis:

(active? (heat-flow water stone container-path)).

The differential predictions supported by this hypothesis for the two scenarios shown in figure 5.8 are obtained by considering the effect of the transformation. The heat flow process that is hypothesised to be active introduces a qualitative proportionality between the rate of the heat flow process and the length of the container path. Since the transformation involves making the length of the container path greater than the length in the original scenario the rate of the hypothesized heat flow process is greater in the transformed scenarios compared to the rate in the original scenario. Accordingly, the hypothesis predicts that the amount of change in the temperatures of the water and stone is greater than the corresponding changes in the original scenario.

The abstract hypothesis:

(active? (heat-flow water vapor vapor-path))

also introduces a qualitative proportionality connecting the rate of the heat flow to the path connecting the water with its vapor. However, the transformation involving the change to the

container path does not affect this qualitative proportionality. Accordingly, this hypothesis predicts that the rates of the hypothesized heat flow process is the same for both scenarios and therefore the amount of the changes to the temperatures of the water and the stone are equal in both scenarios.

5.3.3. Strategies for Experiment Design

5.3.3.1. Elaboration

Figure 5.9 shows the procedure for designing experiments based on elaboration. The procedure involves constructing experiments to measure those quantities whose changes were predicted by a hypothesis and which are easy to measure. The domain knowledge required is the quantities of the domain that are easily measurable in a scenario. Such knowledge can be specified as predicates in the following manner:

```
(easily-measurable (?change amount-of ?liquid)).
```

```
Procedure Elaboration
   For quantity in easily-measurable-quantities do
      Design elaboration experiment to measure quantity
        ;;; involves checking if the quantity is measurable in the scenario and determining which
        ;;; of the permissible values of the quantity are supported by each hypothesis.
   For each elaboration experiment designed do
      Perform each designed experiment
        ;;; involves asking user for the values obtained from the experiment assuming it is
        ;;; performed in the real world. The implementation supports ambiguous and
        ;;; inconclusive experimental results.
      Refute hypotheses that are incompatible with the experimental values
        ;;; involves eliminating those hypotheses none of whose predicted values are compatible
        ;;; with any of experimentally determines values.
```

Figure 5.9 The procedure for designing elaboration experiments.

Example Illustrating Elaboration

Figure 5.10 shows a scenario in which a solution of salt in water is placed in contact with some salt in an open container. The walls of the container are insulated against heat flow. A stone which is at a lower temperature than that of the solution is in contact with the wall of the container. Figure 5.11 shows the initial theory given to the system. The initial theory includes process descriptions for the evaporation of liquids, the dissolving of substances in solutions and the heat flow between objects. This theory fails to explain an observation made in the scenario shown in figure 5.10 – the temperature of the solution is observed to be decreasing and the theory predicts that the

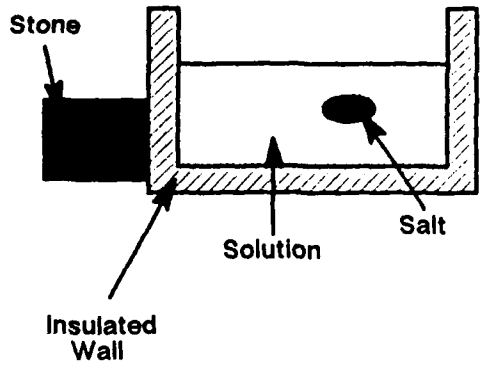


Figure 5.10 A scenario in which a solution of salt in water is placed in contact with some salt in an open container. A stone which is at a lower temperature than that of the solution is in contact with the wall of the container. The wall of the container is insulated against heat flow.

temperature remains constant. Three abstract hypotheses that can explain the unexpected observation are:

H_1: (Causes? (evaporation solution vapor) (decrease (temperature solution)))
H_2: (Causes? (dissolve salt solution) (decrease (temperature solution)))
H_3: (Active? (heat-flow solution stone container-path)).

The easily measurable quantities in the scenario are:

(easily-measurable (?change level ?liquid))
(easily-measurable (?change color ?object)).

All three hypotheses are consistent with the observed decrease in the level of the solution and the constant color of the solution. Therefore, elaboration based on these two quantities does not refute any of the three hypotheses of the evaporation example. Suppose, however that the stone is such that an increase in its temperature causes the color of the stone to become darker. Then, according to the third hypothesis, the color of the stone must become darker. Elaboration will recommend noticing the change in the color and if the color remains constant the third hypothesis is refuted.

```
Evaporation (?liquid ?vapor)
        Individuals
                ?liquid
                ?vapor
        Preconditions
                (open? (container ?liquid))
        Quantity Conditions
        Relations
                (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))
        Influences
                I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
                I+[(amount-of ?vapor), (A (evaporation-rate ?self))]

Dissolve (?solution ?solid)
        Individuals
                ?solution
                ?solid
        Preconditions
                (dissolves? ?solid ?solution)
        Quantity Conditions
        Relations
                (Q+ (dissolve-rate ?self) (contact-area ?solution ?solid))
        Influences
                I-[(amount-of ?solid), (A (dissolve-rate ?self))]
                I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]

Solution (?solution)
        Individuals
                ?solution
        Preconditions
        Quantity Conditions
                (greater-than (A (amount-of (solute-of ?solution))) 0)
        Relations
                (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))
                (Q- (concentration ?solution) (amount-of (solvent-of ?solution)))
                (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution)))

Heat-Flow (?source ?destination ?path)
        Individuals
                ?source
                ?destination
                ?path
        Preconditions
                (heat-aligned? ?path)
        Quantity Conditions
                (greater-than (A (temperature ?source))
                              (A (temperature ?destination)))
        Relations
                (Q+ (heat-flow-rate ?self) (temperature ?source))
                (Q- (heat-flow-rate ?self) (temperature ?destination))
                (Q+ (heat-flow-rate ?self) (cross-sectional-area ?path))
                (Q- (heat-flow-rate ?self) (length ?path))
        Influences
                I-[(temperature ?source), (A (heat-flow-rate ?self))]
                I+[(temperature ?destination), (A (heat-flow-rate ?self))]
```

Figure 5.11 The domain theory for the evaporation example.

5.3.3.2. Discrimination

The procedure for constructing discrimination experiments is shown in figure 5.12. Discrimination requires comparing the predictions of the different hypotheses. It involves selecting a discriminant – a quantity which is measurable and for which some hypotheses predict incompatible values An experiment to measure the discriminant is designed. The hypotheses that predict values that are not consistent with the observed values from the experiment are eliminated.

```
Procedure Discrimination
   Compare the predictions supported by each hypothesis to obtain the set of discriminants
     ;;; involves checking whether at least one hypothesis supports predictions that are
     ;;; incompatible with the predictions supported by some other hypothesis.
   Order the discriminants
     ;;; COAST prefers discriminants that split hypotheses equally.
   Select a number of discriminants
     ;;; The number of discriminants to be selected is a user-supplied parameter
     ;;; of the experiment engine. COAST selects the specified number of best
     ;;; discriminants as established by the above ordering.
   For each selected discriminant do
      Design a discrimination experiment to measure the discriminant
        ;;; involves checking if the discriminant is measurable in the scenario and
        ;;; grouping the hypotheses into sets supporting compatible values of the
        ;;; discriminant.
   For each discrimination experiment designed do
      Perform each designed experiment
        ;;; involves asking user for the values obtained from the experiment assuming it is
        ;;; performed in the real world. The implementation supports ambiguous and
        ;;; inconclusive experimental results.
      Refute hypotheses that are incompatible with the experimental values
        ;;; involves eliminating those hypotheses none of whose predicted values are compatible
        ;;; with any of experimentally determines values.
```

Figure 5.12 The procedure for designing discrimination experiments.

Example Illustrating Discrimination

In the evaporation example, the third hypothesis, which hypothesizes that a heat flow process from the solution to the stone is active, predicts that the temperature of the stone increases. The other two hypotheses predict that the temperature of the stone remains constant. An increase in the temperature of the stone is not compatible with the temperature of the stone remaining constant Therefore, the temperature of the stone can be used as a discriminant and a discrimination experiment is designed to measure the change. If the temperature of the stone is found to be increasing then the first two hypotheses can be eliminated and if the temperature of the stone is found to remain constant the third hypothesis can be eliminated.

5.3.3.3. Transformation

Figure 5.13 describes the procedure for transformation. Transformation involves generating new scenarios and performing elaboration and discrimination on the new scenarios to refute hypotheses. Differential elaboration and discrimination, which involve comparing the behavior of a quantity in two different scenarios, are handled as the discrimination and elaboration of quantities that specify the differential behavior of a property across two scenarios.

```
Procedure Transformation (scenario hypotheses)
    Find the applicable transformation operators for the given scenario and hypotheses
    Select an operator from the above set and apply it to the original scenario
    Select a scenario from the resulting scenarios
```

Figure 5.13 The procedure for transforming scenarios.

The domain knowledge required for transformation is a set of operators that generate new scenarios from a given scenario by manipulating different parameters of the scenario. These operators are general-purpose transformations on scenarios and are indexed by the type of the hypothesis. This is an example of a knowledge-intensive (but not domain specific) strategy used by the experiment engine in preference to the default breadth-first search strategy.

The form of a scenario transformation operator is:

ST-OP(hypothesis, scenario) → scenario.

The operator produces a transformed scenario based on the type of the hypothesis and the input scenario. Nine types of scenario transformation operators have been implemented in COAST:

(1) **Active? Transformation:**

This operator is applicable to hypotheses of the type:

(active? ?process).

The procedure for constructing the transformed scenarios is shown in figure 5.14. The procedure involves 1) determining the manipulable parameters on which the rate of the process hypothesized to be active depends, and 2) constructing scenarios in which the the rate of the process is enhanced or inhibited by changing one of the identified parameters. The ordering of the transformed scenarios is determined by the number of other processes

also hypothesized to be active and whose rates are affected by the transformation. If there are a large number of such hypotheses, the selection of a transformation which affects the rates of many of the processes hypothesized to be active is better than the selection of a transformation that affects the rate of a single process since, in the former case, conducting experiments in the transformed scenario can eliminate many hypotheses simultaneously.

```
Procedure Active?-Transformation (hypothesis scenario)
   Collect the parameters that affect the rate of the process hypothesized to be active
     ;;; Involves examining the qualitative proportionality relations of the process.
   For each parameter do
      If the parameter is manipulable
         then construct a scenario in which the value of the parameter is greater than or less
         than the value in the original scenario
   Order and return the constructed scenarios
     ;;; Ordering involves determining the number of other processes hypothesized to be active
     ;;; and whose rates are affected by the transformation.
```

Figure 5.14 The procedure for transforming scenarios to test the active? hypothesis.

(2) Causes? Transformation:

This transformation is applicable to hypotheses of the type:

(causes? ?process ?observation).

The procedure for constructing the transformed scenarios is shown in figure 5.15. The procedure involves 1) determining the manipulable conditions of the process hypothesized to cause the observation, and 2) constructing a scenario in which one of the conditions is not satisfied. The ordering of the transformed scenarios is influenced by the number of other processes hypothesized to cause the observation and which are affected by the transformation of the condition.

```
Procedure Causes?-Transformation (hypothesis scenario)
   Collect all the conditions of the process that is hypothesized to cause the observation
   For each condition do
      If the condition is manipulable
         then create a scenario in which the condition is not satisfied
   Order and return all the constructed scenarios
     ;;; Ordering involves determining the number of other processes hypothesized to cause
     ;;; the observation and which are affected by the transformation.
```

Figure 5.15 The procedure for transforming scenarios to test the causes? hypothesis

(3) Inactive? Transformation:

This transformation is applicable to hypotheses of the type:

(inactive? ?process).

The procedure for constructing the transformed scenarios is shown in figure 5.16. The procedure is similar to the procedure for constructing scenarios for the *active?* type of hypotheses.

```
Procedure Inactive?-Transformation (hypothesis scenario)
   Collect the parameters that affect the rate of the process hypothesized to be inactive
     ;;; Involves examining the qualitative proportionality relations of the process.
   For each parameter do
      If the parameter is manipulable
         then construct a scenario in which the value of the parameter is greater than or less
         than the value in the original scenario
   Order and return the constructed scenarios
     ;;; Ordering involves determining the number of other processes hypothesized to be
     ;;; inactive and whose rates are affected by the transformation.
```

Figure 5.16 The procedure for transforming scenarios to test the inactive? hypothesis.

(4) Not-Causes? Transformation:

This transformation is applicable to hypotheses of the type:

(not-causes? ?process ?observation).

The procedure for constructing the transformed scenarios is shown in figure 5.17. It is similar to the procedure for constructing scenarios for the *causes?* type of hypotheses.

```
Procedure Not-causes?-Transformation (hypothesis scenario)
   Collect all the conditions of the process that is hypothesized not to cause the observation
   For each condition do
      If the condition is manipulable
         then create a scenario in which the condition is not satisfied
   Order and return all the constructed scenarios
     ;;; Ordering involves determining the number of other processes hypothesized not to cause
     ;;; the observation and which are affected by the transformation.
```

Figure 5.17 The procedure for transforming scenarios to test the not-causes? hypothesis.

(5) Equals? Transformation:

This transformation is applicable to hypotheses of the type:

(equals? ?change1 ?change2).

The procedure for constructing the transformed scenarios is shown in figure 5.18. The procedure involves 1) determining the manipulable parameters on which the rate of a process that causes the known change depends and 2) constructing scenarios in which the the rate of the process is enhanced or inhibited by changing one of the identified parameters.

```
Procedure Equals?-Transformation (hypothesis scenario)
   Collect the parameters that affect the rate of a process that causes the known change
     ;;; involves examining the qualitative proportionality relations of the process.
   For each parameter do
      If the parameter is manipulable
         then construct a scenario in which the value of the parameter is greater than or less
         than the value in the original scenario
   Return the constructed scenarios
```

Figure 5.18 The procedure for transforming scenarios to test the equals? hypothesis.

(6) Dominates? Transformation:

This transformation is applicable to hypotheses of the type:

(dominates? ?change1 ?change2).

The procedure for constructing the transformed scenarios is shown in figure 5.19. It is similar to the procedure for constructing scenarios for the *equals?* type of hypotheses.

```
Procedure Dominates?-Transformation (hypothesis scenario)
   Collect the parameters that affect the rate of a process that causes the known change
     ;;; involves examining the qualitative proportionality relations of the process.
   For each parameter do
      If the parameter is manipulable
         then construct a scenario in which the value of the parameter is greater than or less
         than the value in the original scenario
   Return the constructed scenarios
```

Figure 5.19 The procedure for transforming scenarios to test the dominates? hypothesis.

(7) Revised Relation Transformation:

This transformation is applicable to theories that are obtained by the revision of a qualitative proportionality relation of a process. The procedure for constructing the transformed scenarios is shown in figure 5.20. The procedure involves 1) identifying an inactive process

that influences the independent quantity of the revised relation, and 2) constructing a new scenario in which the failed conditions of the process are satisfied.

```
Procedure Revised-Relation-Transformation (hypothesis scenario)
   Find all the inactive processes that influence the independent quantity of the revised relation
   For each inactive process do
      Collect all the failed conditions of the process
      If all the failed conditions are manipulable
         then create a scenario in which the failed conditions are satisfied
   Order and return all the constructed scenarios
     ;;; Ordering involves determining the number of other revised relations that are affected
     ;;; by the transformation.
```

Figure 5.20 The procedure for transforming scenarios to test a theory obtained by revising a relation of a process.

(8) Revised Condition Transformation:

This transformation is applicable to theories that are obtained by the revision of a quantity condition of a process. The procedure for constructing the transformed scenarios is shown in figure 5.21. The procedure involves 1) identifying an inactive process that influences one of the quantities of the revised quantity condition, and 2) constructing a new scenario in which the failed conditions of the process are satisfied.

```
Procedure Revised-Condition-Transformation (hypothesis scenario)
   Find all the inactive processes that influence one of the quantities of the revised quantity
   condition
   For each inactive process do
      Collect all the failed conditions of the process
      If all the failed conditions are manipulable
         then create a scenario in which the failed conditions are satisfied
   Order and return all the constructed scenarios
     ;;; Ordering involves determining the number of other revised quantity conditions
     ;;; that are affected by the transformation.
```

Figure 5.21 The procedure for transforming scenarios to test a theory obtained by revising a quantity condition of a process.

(9) Unobservable Effects Transformation:

This transformation is applied to hypotheses that entail unobservable effects. The procedure for the construction of scenarios is shown in figure 5.22. The procedure involves: 1) identifying an inactive process which indirectly influences one of the unobservable effects, 2) constructing a new scenario in which the failed conditions of the inactive process are

satisfied, and 3) checking whether the changes due to the unobservable effects are measurable in the constructed scenario.

Procedure Unobservable–Effects–Transformation (hypothesis scenario)
  Find all the inactive processes that indirectly influence one of the unobservable effects entailed by the hypothesis
  For each inactive process do
    Collect all the failed conditions of the process
    If all the failed conditions are manipulable
      then create a scenario in which the failed conditions are satisfied
  Select a scenario from those constructed scenarios in which the changes due to the unobservable effect are measurable

Figure 5.22 The procedure for transforming scenarios to test the unobservable effects entailed by a hypothesis.

Example Illustrating Transformation

In the evaporation example, the first hypothesis states that the active process evaporation of the solution causes the temperature of the solution to decrease. According to the first transformation operator described above, the original scenario can be transformed to a new scenario in which the evaporation process is not active because a precondition which can be manipulated in the scenario has been defeated. In the scenario, the container can be closed to prevent evaporation of the solution. Figure 5.23 depicts such a scenario. According to the first hypothesis, the temperature of

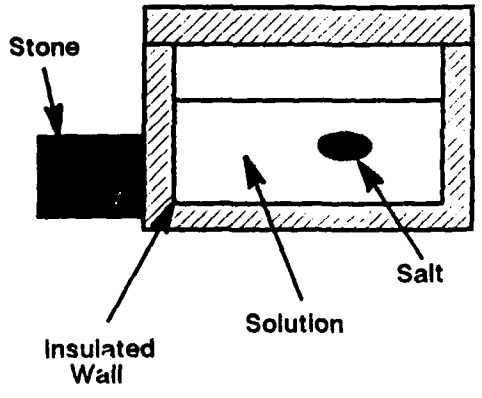


Figure 5.23 A scenario obtained by transforming the scenario shown in figure 5.10. The container is closed to prevent evaporation of the solution.

the solution remains constant in the transformed scenario. However, the other two hypotheses are

not affected by the transformation and they predict that the temperature of the solution decreases. Discrimination in the transformed scenario will recommend measuring the temperature of the solution. If the temperature of the solution is found to decrease then the first hypothesis can be eliminated and if it is found to remain constant the other two hypotheses can be eliminated.

As another example of transformation consider the third hypothesis in the evaporation example. This hypothesis states that a heat flow process from the solution to the stone is active is responsible for the observed decrease in the temperature of the solution. The second transformation operator is applicable. The rate of the heat flow process depends on the geometry of the path of the heat flow – it is directly proportional to the cross–sectional area of the path and inversely proportional to the length of the path. One of the scenarios obtained by applying the operator is shown in figure 5.24. In

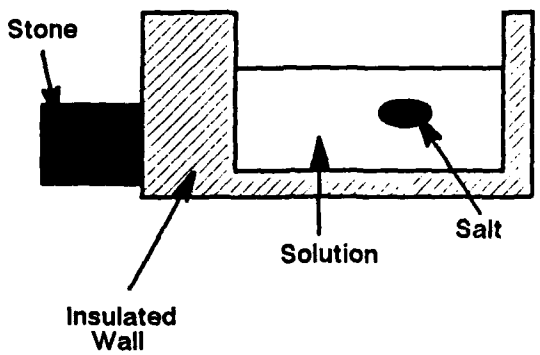


Figure 5.24 A scenario obtained by transforming the scenario shown in figure 5.10. The wall of the container in contact with the stone is extended in increase the length of the heat flow path.

this transformed scenario the wall of the container which is the path of the hypothesized heat flow is extended. The third hypothesis predicts that the temperature of the solution decreases at a much slower rate in the transformed scenario as compared to the decrease in the original scenario. However, this transformation does not affect the other hypotheses. Therefore, according to the first and second hypotheses the decrease in the temperature of the solution is the same in both the scenarios. Differential discrimination will recommend comparing the change in the temperature of the solution in the two scenarios.

5.4. Evaluation of Experimentation–based Hypothesis Refutation

This section defines four criteria for evaluating an experiment design system: *efficacy*, *efficiency*, tolerance of unavailable data and feasibility. Experimentation–based hypothesis refutation, the method for designing experiments, and its implementation in COAST are analyzed with respect to each of the four criteria.

5.4.1. Efficacy

The efficacy of an experiment design system is defined to be the system's ability to find an experiment to refute a hypothesis if the experiment exists. The efficacy of experimentation–based hypothesis refutation is determined by:

1) The set of transformation operators supplied to the system. If a hypothesis can be tested only by transforming the original scenario and the operator that is required to transform the scenario is missing then the system cannot design an experiment to test the hypothesis.

2) The set of predictions obtained from the inference engine. If a hypothesis can be tested only by measuring a particular quantity and if the inference engine cannot compute the behavior of the quantity predicted by the hypothesis then the system cannot test the hypothesis.

3) The set of predicates that specify the quantities that can be measured, the values of each quantity that are incompatible, and the parameters of the scenario that can be manipulated. If a hypothesis can be tested only by measuring a particular quantity and if the system does not know that the quantity can be measured or that its values are incompatible then the system cannot design an experiment to measure the quantity.

If COAST is equipped with complete sets of transformation operators, predictions, and predicates then it eventually finds the experiment through a brute force search in which experiments are designed for each measurable quantity in every scenario that can be constructed through transformation. Efficacy can be sacrificed for efficiency in COAST by 1) using a heuristic search or a beam search for the transformation of scenarios or by generating scenarios using knowledge about the hypotheses 2) investigating only a selected set of transformed scenarios 3) examining only a selected set of predictions from the inference engine, and 4) designing only a selected set of elaboration and discrimination experiments.

5.4.2. Efficiency

An experiment design system is efficient if it designs the least number of experiments to measure quantities and constructs the least number of transformed scenarios required to test a hypothesis. Experimentation-based hypothesis refutation involves two decision points that affect its efficiency:

1) Selection of a transformation operator:

   A number of transformation operators may be applicable resulting in many different scenarios. The selection of a scenario can affect the efficiency of the system. If the system selects a scenario that results in experiments that do not refute any of the hypotheses then the effort involved in transforming and exploring the scenario is wasted.

2) Selection of a discriminant:

   A number of discriminants may be available for discrimination experiments. The selection of discriminants can affect the efficiency of the system. For example, if the system selects a discriminant that groups a large number of hypotheses into two sets – one set consisting of one hypothesis and the other consisting of the rest – then the experiment can result in the refutation of only the set containing the single hypothesis. In contrast, another discriminant that splits the hypotheses equally into two sets yields an experiment that can result in the refutation of half of the original hypotheses.

COAST improves its efficiency by 1) employing a knowledge-intensive transformation scheme based on the hypotheses to guide the transformation process and 2) ordering discriminants based on their ability to split the hypotheses as evenly as possible.

An additional factor that governs the efficiency of the experimentation system is the cost of running the experiments. Though an experiment can be easy to design, there is no guarantee that it will be cost effective to run the experiment. The investigation of the tradeoffs in the costs of designing and performing the experiments is a topic of future research and is not addressed by the current research.

5.4.3. Tolerance of Unavailable Data

An experiment design system is tolerant of unavailable data if it can construct alternate experiments if the current experiment fails to yield any information. An experiment fails if it is inconclusive – the experimentally determined values are the values known prior to the experiment – or if it is so

ambiguous that the experimentally determined values are still compatible with all the hypotheses. Experimentation-based hypothesis refutation is made tolerant of unavailable data by incorporating an element of redundancy in the design of experiments. Note that this redundancy is introduced at the expense of efficiency because if the original experiment succeeds then the system has unnecessarily expended resources in constructing the additional experiments. According to the procedure described in figure 5.5 there are two decision points at which redundancy can be introduced 1) the selection of a transformation operator and 2) the selection of a discriminant.

COAST requires the user to specify an upper bound on the number of discriminants selected for discrimination experiments in each scenario and the number of scenarios investigated. Within these limits, COAST introduces redundancy in:

1) The selection of a transformation operator:

   The system is capable of selecting alternate means of transforming the scenario if the initial transformation fails. For example, while testing if a process is active, if varying a parameter of the original scenario that affects the rate of the process fails to produce conclusive evidence then the system makes additional transformations that involve varying other parameters of the scenario that also affect the rate of the process.

2) The selection of a discriminant:

   The system is capable of selecting alternate discriminants if the the discrimination experiment based on the original discriminant is inconclusive or too ambiguous. For example, to test if a flow is occurring or not, if the measurement of the change of the amount of the liquid at the source is inconclusive then the system measures the change in the amount of the liquid at the destination also.

5.4.4. Feasibility

The experiment design system is practical if it proposes only those experiments that are feasible in the real world. Each of the three strategies used by experimentation-based hypothesis refutation employ domain knowledge to determine if a quantity can be measured or manipulated in a particular scenario. Therefore, experimentation-based hypothesis refutation does not design experiments to measure quantities that cannot be measured or that requires manipulating a parameter that cannot be accessed or changed.

COAST ensures that the experiments are feasible by using domain knowledge to ascertain that the experimental measurements and manipulations are practicable. The domain knowledge includes predicates that specify the quantities of the domain that are measurable, the quantities that are easily measurable, the values of each quantity that are not compatible, and the parameters of a scenario that are manipulable. Elaboration designs experiments to measure any quantity that is easily measurable. Discrimination designs experiments to measure only those quantities that are measurable. Transformation manipulates only those parameters that are given to be manipulable in the required manner. For example, while constructing scenarios to vary the rate of the process, the system checks whether the parameter to be varied, such as the dial on the gas burner, is manipulable in the required manner.

5.5. Discussion

A recent discovery system, IDS [Nordhausen87], also proposes experiments to gather data. However, the experiments generated by IDS are exploratory in nature and, unlike COAST's experiments, are not directed towards the refutation of well-formed hypotheses. Carbonell and Gil [Carbonell87] have recently proposed a system that learns new preconditions and postconditions for STRIPS-like operators by experimentation. Their experimentation techniques involve comparing states of the world to identify differences and to identify operators with a missing postcondition. COAST's experimentation techniques are based on methods that compare the predictions of hypotheses and is more widely applicable.

A method to cope with multiple hypotheses was outlined in this chapter. The approach, called *experimentation-based hypothesis refutation*, can be invoked by a theory revision system confronted with multiple hypotheses that propose revisions to the original theory. When invoked, the system identifies measurements which would serve to disambiguate among the postulated revisions. The system then proposes experiments by which these measurements can be obtained. A major portion of the research contribution addressed the problem of *experiment design*.

CHAPTER 6

EXEMPLAR-BASED THEORY REJECTION

6.1. Introduction

Depending on the failure, many different types of revisions to the initial theory are possible – components of the theory may be deleted, new components may be added, components may be negated, the scope of the components may be widened or narrowed. However, there is a problem with the revised theories generated by the hypothesis generator described in chapter 4. The initial theory is used by the problem solving system to explain observations in a number of scenarios. Though the revised theories explain the observations in the scenario in which the failure is encountered, they will not necessarily explain the observations in the scenarios previously encountered by the system that were successfully explained by the initial theory. This problem is particularly acute for revised theories obtained by making changes such as deleting a component or negating a component. The component that is deleted or negated may have been essential to the explanation of some previously observed phenomenon and the deletion or negation of the component can result in a revised theory that predicts that the phenomenon does not occur or that the inverse phenomenon occurs.

It is important that the revised theories explain those scenarios that were explained by the original theory. Otherwise, the theory revision system may end up oscillating between two incorrect theories and may never converge on a theory that can explain all the observations. For example, a failure may lead to a revised theory, T_1, that involves deleting a component from the initial theory, T_0. However, the next failure may be such that a revised theory, T_2, that incorporates the deleted component, can explain the observation. The two theories, T_0 and T_2, are identical. If the two scenarios are repeated, then the system will oscillate between the two theories and will fail to select a theory, T_3, that explains the observations in both the scenarios. Notice that such a problem is not

due to theories obtained by deleting components alone. Revised theories generated by inverting, negating, narrowing or widening the scope, or even adding components can also lead to this problem.

The role played by the components of the initial theory that were revised has to be identified in order to insure that the same role is retained by the revised components of the revised theory or is maintained by other components of the revised theory. For example, the components of the theory that were revised may have been instrumental in explaining some observations in a scenario previously encountered by the system. This imposes a constraint on the theory that is produced by modifying these components – the revised theory must also explain these observations. The revisions may be such that the old explanations are not affected. Or the revised theory may construct entirely new explanations that do not require the revised components. Or the revised theory may incorporate the revised components into the explanations. The important point is that the past observations must be explainable by the revised theory – it does not matter if the same components of the theory are used or if the explanation is different.

Exemplar–based theory rejection is a method for testing revised theories. The method involves collecting and organizing the system's history of explained observations and using these to test each revised theory by verifying that the theory explains these observations. Exemplar–based theory rejection functions like a filter on the input theories. It filters out those theories that cannot explain previous observations. The output can consist of many theories if each theory can explain the previous observations.

The basic requirement of exemplar–based theory rejection is a rich database of the scenarios encountered and successfully explained by the theory. Some of the interesting issues that must be addressed by the method are: Are all the observations in each scenario encountered by the system stored or are only a few? If only a few of the observations are selected then what are the criteria according to which the selection is made? How is the database of past observations built and maintained as new scenarios are encountered? Does every observation in the database have to be re–explained to test a proposed theory or only a selected subset? If a selected subset is sufficient then which observations are selected to be re–explained? If a number of revised theories can successfully explain the past observations, how is the database of observations for each revised theory constructed from the old database?

As an example, consider an initial theory that consists of process definitions for heat flow and evaporation. Suppose the evaporation process definition is initially incomplete since it does not affect the temperature of the evaporating liquid. Figure 6.1a shows a scenario in which water is

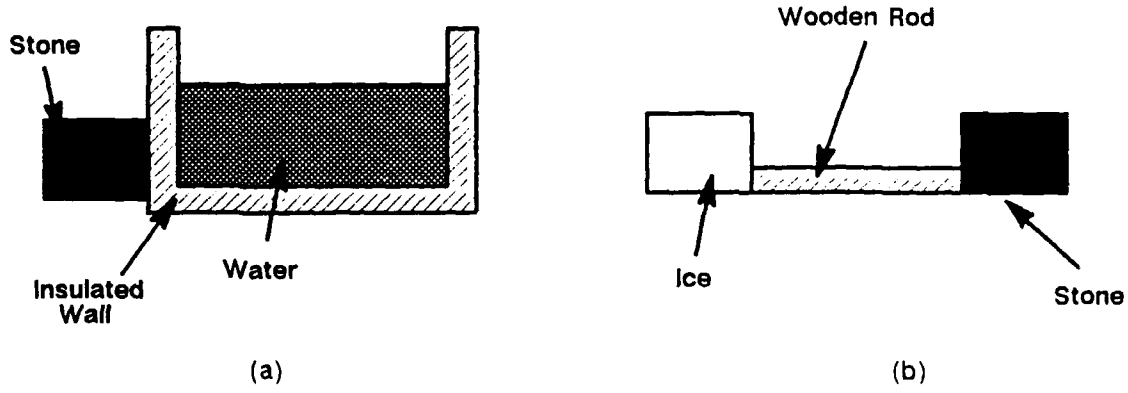


Figure 6.1 (a) A scenario in which water is placed in an open container. The wall of the container is insulated against heat flow. A stone at a much lower temperature that that of water is in contact with the container's wall. (b) An ice block is connected to a stone by a wooden rod which does not conduct heat.

placed in an open container. The walls of the container are insulated against heat flow. One of the walls touches a stone which is at a lower temperature than the water in the container. The theory predicts that the temperature of water is constant in the scenario. However, the observation made from the scenario is that the temperature of water is decreasing. A revision to the theory that can explain this observation is the deletion of the precondition requiring the path of the heat flow to be conducting. Consequently, the revised theory predicts that a heat flow from the water in the container to the stone occurs through the connecting wall even though the path is specified to be insulating. This heat flow process can explain the observed decrease in the temperature of water. However, the revised theory cannot explain the observations of a scenario satisfactorily accounted for earlier by the system (figure 6.1b). In this scenario, an ice block and a stone are connected by a wooden rod which does not conduct heat. The temperatures of the ice block and the stone were observed to be constant. The revised theory predicts that the heat flow process occurs because the ice block and the stone are connected and therefore predicts that the temperatures of the ice block and the stone are decreasing and increasing respectively. Since this is not consistent with the observations made in the scenario the revised theory is incorrect and is rejected. In this example,

the revision of a precondition of the heat flow process resulted in the invalidation of the explanations for the observations made in an earlier scenario (figure 6.1b).

The next section presents a theoretical description of exemplar-based theory rejection. The third section describes an implementation of the method for theories represented in Qualitative Process theory. The fourth section presents a theoretical analysis of the method. The last section discusses work that is related to exemplar-based theory rejection and presents a summary of the chapter.

6.2. Exemplar-based Theory Rejection

The method consists of two steps:

1) Representing the system's history of observed phenomena: This involves identifying and collecting examples that illustrate how specific components of the theory are used to explain observed behavior in scenarios.

2) Using these examples to refute revised theories: When a revised theory is proposed, it is tested to verify that it explains the observations in the examples that were collected. This ensures that the revisions to the theory does not result in a theory that makes predictions that are inconsistent with the earlier observed behavior.

6.2.1. Exemplars and Prototypes

An *exemplar* is an example that illustrates the use of components of the theory to explain an observation in a scenario. It consists of four pieces of information: 1) the *observation* that is explained using the theory 2) the *layout of the scenario* in which the observation is made 3) the *explanation* for the observation (constructed by the theory) and 4) the *components* of the theory that were used to construct the explanation.

For a given theory, there is a set of exemplars, called an *exemplar space*, corresponding to the theory, that serves to exemplify the different components of the theory (figure 6.2). An exemplar may simultaneously serve as an example for many different components of the theory. Similarly, a component of the theory may be exemplified by many different exemplars.

A *prototype* for a component is an exemplar which contains information that is minimally sufficient to exemplify the component of the theory. It is an exemplar that satisfies the following two constraints:

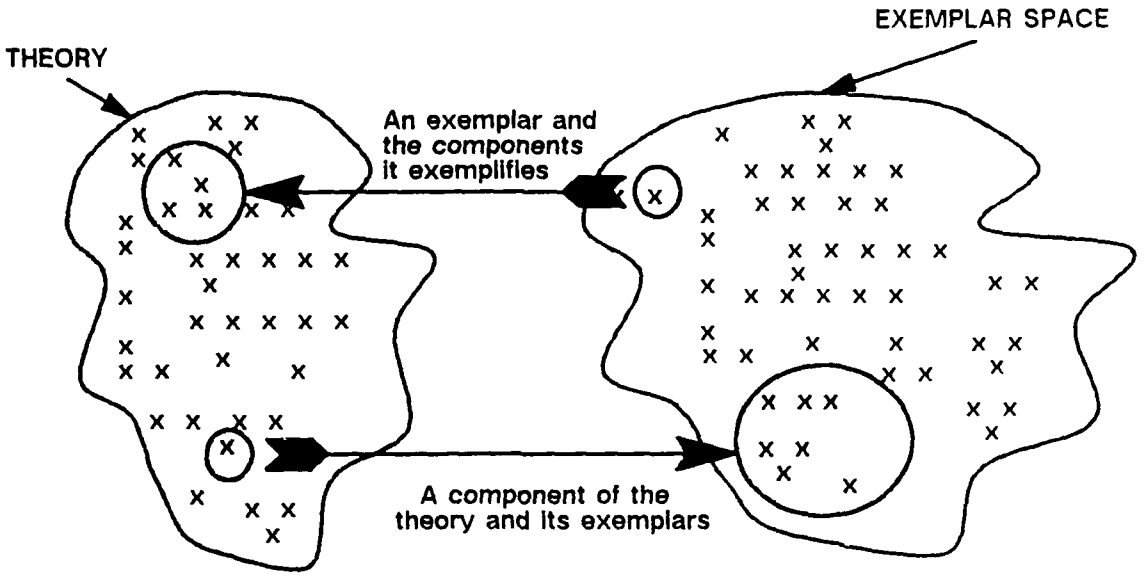


Figure 6.2 The relationship between the theory components and the space of exemplars. A component of the theory can be exemplified by a number of exemplars. An exemplar can exemplify a number of components.

1) The exemplar scenario is simpler than the scenarios of all the other exemplars for the component according to a user-defined metric for comparing scenarios.

2) The exemplar explanation is simpler than the explanations of all the other exemplars for the component that have the scenario identified in step 1 according to a user-defined metric for comparing explanations.

An exemplar that is the prototype for one component of the theory need not be the prototype for the other components of the theory that it also exemplifies.

Determining the prototype of a component requires comparing scenarios and explanations. Measures for the simplicity of scenarios and explanations may be defined based on the representation adopted by the implementation. For example, a scenario that has more objects or compound objects such as solutions may be considered less simple than one that has fewer objects or simpler objects such as solvents. An explanation with many links, that is, a long explanation, may be considered less simple than one with fewer links. If the metrics used provide a

unique simplest explanation and a unique simplest scenario then the prototype for a component will also be unique. Otherwise, a component can have many prototypes.

6.2.2. Forming the Exemplar Space

An exemplar space consists of exemplars for the components of the theory. It is formed incrementally by creating and adding exemplars as the system encounters new scenarios and uses the theory to explain the observations made in the scenarios. An exemplar is created whenever the system successfully explains, based on its domain theory, an observation in a given scenario. The steps involved in the creation of exemplars are:

Explanation Construction: The theory is used to construct an explanation for the observed phenomenon encountered in the scenario.

Identifying Components: The components of the theory that are used to construct the explanation are identified and collected.

Creating an Exemplar: An exemplar that consists of the collected components, the observation, the scenario in which the observation is made and the explanation for the observation is created.

Though the exemplar is created, it need not necessarily be added to the exemplar space corresponding to the theory. For a normal system, the theory will be successfully used to explain a large number of scenarios and each scenario can have a large number of observations that have to be explained. If each exemplar corresponding to every observation in every scenario successfully explained by the theory is added to the exemplar space, the exemplar space can become enormous. Even if such an enormous space can be constructed, it will not be very useful for testing the proposed theories. A large number of exemplars will have to be re-explained for each proposed theory and this may be very expensive. Thus, there are two reasons for adding exemplars to the exemplar space selectively – 1) for a system of reasonable complexity which handles a large number of scenarios, the space of possible exemplars is very large and it may not be possible to construct such exemplar spaces, and 2) in order to efficiently test each proposed theory, it is better to have small exemplar spaces so that only a few exemplars have to be re-explained. The trade-offs involved in restricting the exemplar space in this manner are examined in section 4.

The newly created exemplar is added to the existing exemplar space only if it serves to exemplify components of the theory better than the existing exemplars in the space. There are two criteria used to determine the utility of the new exemplar:

(1) **Exemplar Threshold Criterion:**

A threshold of upto n exemplars for each type of exemplar for a component is established. If the new exemplar is within the threshold limit for any of the components that it exemplifies then it is added to the exemplar space. It is necessary to have more than one exemplar as the threshold limit since the revisions may be such that some exemplars for some components may be withdrawn during the re-explanation phase.

(2) **Prototype Criterion:**

It is a prototype for one of the components of the theory that it exemplifies.

Notice that these criteria allow for any number of exemplars for a particular component of the theory. For example, each new scenario may result in an exemplar for a particular component. The exemplar may be such that it is simpler than the existing exemplars for the component and is, therefore, added to the exemplar space. This may result in large exemplar spaces that are difficult to manipulate. Therefore, a notion of *minimality* of exemplar spaces is defined to prevent the exemplar space from growing excessively large. An exemplar space is *minimal* if deleting an exemplar from the space will result in the violation of one of the two criteria described above. Using this notion of minimality, the exemplar space can be dynamically pruned either when it exceeds a threshold number of exemplars in the space or when a new exemplar is added to the space. Note that according to this definition of minimality, there may be many exemplar spaces that are equally minimal.

The prototype criterion results in the retention of simpler exemplars for the components of the theory. The selection of the simplicity of the explanations and the scenarios as the criterion for determining the prototypicality of an exemplar is motivated by the need to minimize the cost of the re-explanation of the exemplar observations during the testing of the revised theories. However, in many cases, it can be advantageous to retain the more complex exemplars. First, the retention of complex exemplars may result in a more compact exemplar space which consists of a few complex exemplars that serve to exemplify all the components of the theory. In such a case, the

cost of re-explanation of a few complex exemplars for this type of exemplar space may be smaller or comparable to the cost of re-explanation of many simple exemplars for an exemplar space with a large number of simple exemplars. Second, simpler exemplars may miss many of the subtleties of the interaction of the components of the theory that arise only in the complex scenarios. Consequently, simpler exemplars are less useful in testing revised theories. One solution to the problem of deciding whether to retain simple exemplars or complex exemplars is to retain a range of exemplars of varying complexity for each component of the theory. An alternate solution is to retain exemplars based on *typicality* rather than simplicity. However, the notion of typicality is much harder to define. The current research has not investigated the different trade-offs involved in the selection of different criteria to determine prototypical exemplars. This is an important topic of future research.

6.2.3. Using the Exemplar Space

The exemplar space corresponding to a theory can be used to test revised theories. There are three steps to test each revised theory:

Exemplar Retrieval:

The exemplars that have to be re-explained by the revised theory are retrieved. One possibility is to test the revised theory by re-explaining every exemplar observation in the exemplar space. However, this can be very expensive if the exemplar space is large. Also, this may be unnecessary if the revisions to the theory are such that only a small fraction of the exemplars are affected by the changes. This is particularly true if the revisions are localized to a small number of components of the theory (figure 6.3). Therefore, only those exemplars whose explanations might be invalidated due to the revisions are retrieved. The procedure for determining which explanations are affected depends on the representation of the theory. For example, if a component of the old theory is deleted, then the exemplars whose explanations use that component are retrieved.

Explanation Reconstruction:

For each retrieved exemplar, an explanation for the exemplar observation, based on the proposed revised theory, is constructed. If for any of the retrieved exemplars, the explanation cannot be constructed, then the proposed theory is rejected.

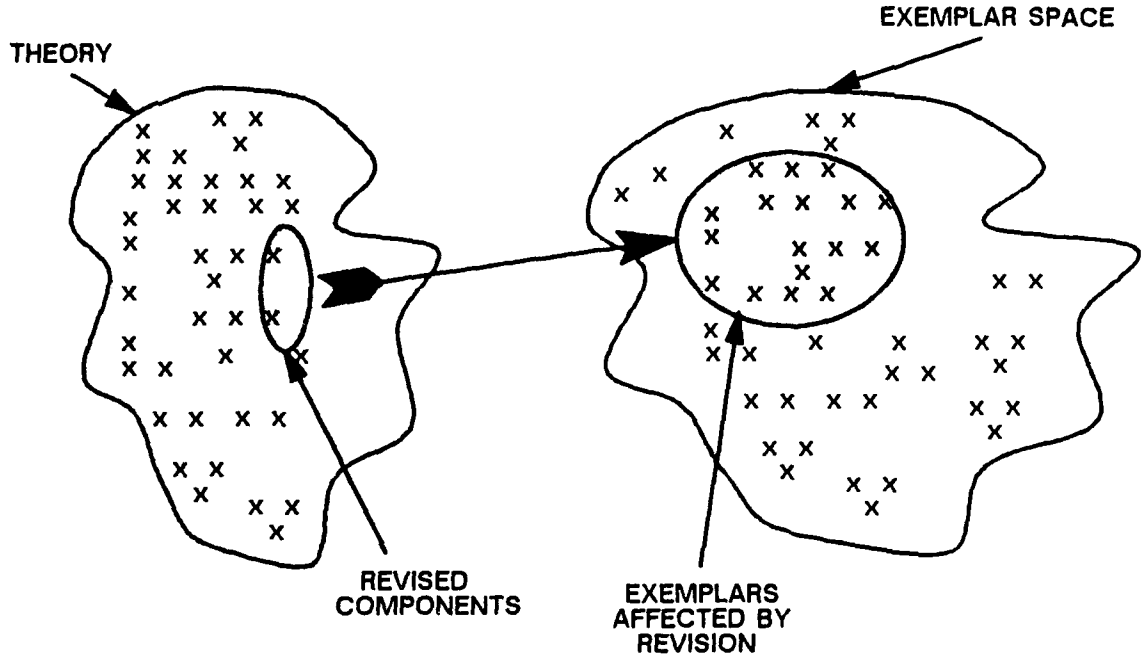


Figure 6.3 The revisions to the theory are such that only the exemplars in the shaded region are affected. The revised theory is tested by re-explaining the observations of these exemplars.

Forming a New Exemplar Space:

If the revised theory can re-explain the observations of all of the retrieved exemplars, then it satisfies the test. An exemplar space corresponding to this theory has to be constructed so that future revisions to this new theory can be tested by the exemplar-based theory rejection method. The exemplar space for the revised theory consists of the union of all the redone exemplars and the exemplars from the old space that were not retrieved in the first step. There may be many proposed theories remaining that satisfy the exemplar re-explanation requirement. Since each of these theories may have been formed by different revisions to different components of the initial theory, each has its own distinct exemplar space (though it may share exemplars with other theories if it shares components).

6.3. Exemplar-based Theory Rejection in COAST

This section describes an implementation of exemplar-based theory rejection that is applicable to theories that are represented in Qualitative Process theory. Such theories can be decomposed into

components with specific functionality such as the preconditions of a process and the influences of a process. The theory consists of a description of the processes in a given domain. The components of the theory are the pieces of information in each process – individuals, preconditions, quantity conditions, relations and influences.

The exemplar-based theory rejection system accepts a set of revised theories represented in Qualitative Process theory as input. Each of these theories provides an explanation for the observations that invoked theory revision. The system tests each of these theories and rejects those that are inconsistent with the past observations of the system as represented by the exemplars. It returns a subset of the input revised theories. Each theory in the subset provides explanations for the exemplar observations and has a distinct exemplar space.

In this section, first, the types of exemplars that are required to test proposed revisions are described. Next, the incremental formation of exemplar spaces is described. Finally, the use of the exemplars in the exemplar space to test proposed theories is described.

6.3.1. Requirements of an Exemplar Space

This section analyzes the types of exemplars required to effectively test proposed revisions to the theory. The type of exemplar used to test a proposed revision depends on the type of revision (add, delete, invert etc) and the components to which it is applied (precondition, influence etc). The components of the process can be grouped into three classes: individuals, conditions and effects.

Revising an individual:

The revision of an individual is secondary and is based on revising a condition or an effect of the process. For example, if a new effect, that involves an object that is not initially an individual of the process, is proposed, then a secondary revision involving the addition of this object to the individuals of the process is generated. Testing the primary revision will automatically test this revision too. Therefore, this revision does not impose any additional constraints on the types of exemplars that are required.

Revising a condition:

If the revision is such that conditions of the process are less constrained (for example, deleting a precondition) and the process becomes active in additional scenarios, then all the

previous scenarios in which the process is inactive due to the failure of the revised condition are affected because the revised conditions may have been satisfied. Therefore, to test the revised theory, the system needs examples of such scenarios and has to re-explain the observed changes in these scenarios. If the revision is such that additional constraints are imposed on the conditions of the process (adding a precondition) and it is inactive in additional scenarios, then all the previous instances of the process being active are affected because the additional constraints may not have been satisfied in these scenarios. Therefore, the system needs examples of scenarios in which the process is active and has to re-explain those observations that depend on the process being active.

Revising an effect:

If the revisions are such that there are more observed changes in scenarios in which the process is active (for example, adding an influence), then the explanations for the observations in all the scenarios in which the process is active are affected because the additional effects may invalidate or change previous explanations. To test this type of revision the system needs examples of scenarios in which the process is active. If the revisions are such that there are less observed changes when the process is active (for example, deleting an influence), then all the previous explanations in which the component is used are affected. Therefore, the system needs examples of such observations to test if the revised theory can re-explain them.

To summarize, the system needs exemplars of:

1) An active process, that is, all the conditions of the process are satisfied.

2) A condition of a process not satisfied and the other conditions satisfied. This illustrates why the particular condition is needed.

3) An effect of a process used in the explanation of an observation.

6.3.2. Forming the Exemplar Space

An exemplar is created whenever the system successfully uses its domain theory to explain an observation in a given scenario. Explanations for observed changes produce exemplars of active processes and exemplars illustrating the use of the effects of processes (exemplars of type 1 and 3

in the above section). Explanations for quantities that are observed to remain constant because a process is not active due to a failed condition yield exemplars of type 2 that illustrate why a condition is required.

6.3.2.1. Examples

Examples involving the evaporation of liquids, the dissolving of substances in solutions and the flow of heat between objects are used to illustrate exemplars and the incremental formation of exemplar spaces. Figure 6.4 shows the domain theory used in the examples. It consists of definitions for evaporation of liquids, dissolving of substances in solutions, solutions and the flow of heat between objects.

6.3.2.2. Evaporation of a Solution of Salt

Figure 6.5 describes a scenario in which a solution of salt is placed in an open container. Figure 6.6 shows the behavior of the scenario that is predicted by the theory described in figure 6.4. Evaporation of the salt solution is the only active process. The changes predicted are a decrease in the amount of the salt solution and an increase in the amount of vapor. The explanations for these changes are shown in the description of the behavior.

Two exemplars are formed in this example – one for each observed change. The exemplars are shown in figure 6.7. The first exemplar (for the observed increase in the amount of vapor) exemplifies two components of the evaporation process – the precondition *open?* and the influence affecting the amount of the vapor. The second exemplar (for the observed decrease in the amount of the salt solution) also exemplifies two components of the evaporation process – the precondition *open?* and the influence affecting the amount of the salt solution. Both the exemplars are added to the exemplar space because they satisfy the required criteria. The exemplar space after this example consists of only these two exemplars. Only three components of the theory are exemplified – the two influences and the precondition of the evaporation process. Figure 6.8 shows the relationship between the theory space and the exemplar space after this example. Notice that no exemplars are formed for the solution since none of the relations in the definition of the solution are used to explain either of the observed changes.

```
Evaporation (?liquid ?vapor)                                              C
        Individuals
                ?liquid                                                   C1
                ?vapor                                                    C2
        Preconditions
                (open? (container ?liquid))                               C3
        Quantity Conditions
        Relations
                (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))  C4
        Influences
                I-[(amount-of ?liquid), (A (evaporation-rate ?self))]     C5
                I+[(amount-of ?vapor), (A (evaporation-rate ?self))]      C6

Dissolve (?solution ?solid)
        Individuals
                ?solution                                                 C7
                ?solid                                                    C8
        Preconditions
                (dissolves? ?solid ?solution)                             C9
        Quantity Conditions
        Relations
                (Q+ (dissolve-rate ?self) (contact-area ?solution ?solid))  C10
        Influences
                I-[(amount-of ?solid), (A (dissolve-rate ?self))]         C11
                I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]  C12

Solution (?solution)
        Individuals
                ?solution                                                 C13
        Preconditions
        Quantity Conditions
                (greater-than (A (amount-of (solute-of ?solution))) 0)    C14
        Relations
                (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))  C15
                (Q- (concentration ?solution) (amount-of (solvent-of ?solution)))  C16
                (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution)))  C17

Heat-Flow (?source ?destination ?path)
        Individuals
                ?source                                                   C18
                ?destination                                              C19
                ?path                                                     C20
        Preconditions
                (heat-aligned? ?path)                                     C21
        Quantity Conditions
                (greater-than (A (temperature ?source))
                              (A (temperature ?destination)))             C22
        Relations
                (Q+ (heat-flow-rate ?self) (temperature ?source))         C23
                (Q- (heat-flow-rate ?self) (temperature ?destination))    C24
                (Q+ (heat-flow-rate ?self) (cross-sectional-area ?path))  C25
                (Q- (heat-flow-rate ?self) (length ?path))                C26
        Influences
                I-[(temperature ?source), (A (heat-flow-rate ?self))]     C27
                I+[(temperature ?destination), (A (heat-flow-rate ?self))]  C28
```

Figure 6.4 The domain theory used in the following examples. It consists of definitions for evaporation of liquids, dissolving of substances in solutions, solutions and heat flow between objects.

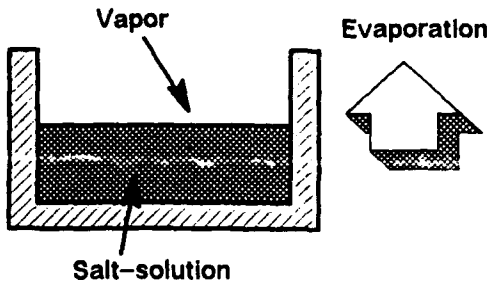


Figure 6.5 A scenario in which a salt solution is placed in an open container in contact with its vapor.

```
Behavior1:
   Theory: <Evaporation> <Heat-Flow> <Dissolve> <Solution>
   Scenario: <salt-solution-evaporation-scenario>
   Active Processes:
            (Evaporation salt-solution vapor)  (solution salt-solution)
   Inactive Processes:
   Predicted Changes:
            Increase (amount-of vapor)
            Decrease (amount-of salt-solution)
   Explanations:
      (Increase (amount-of vapor))
         I+[(amount-of vapor),
                  (A (evaporation-rate (evaporation salt-solution vapor)))]
            (active (evaporation salt-solution vapor))
               (open? (container salt-solution))
      (decrease (amount-of salt-solution))
         I-[(amount-of salt-solution),
                  (A (evaporation-rate (evaporation salt-solution vapor)))]
            (active (evaporation salt-solution vapor))
               (open? (container salt-solution))
```

Figure 6.6 The behavior of the scenario described in figure 6.5.

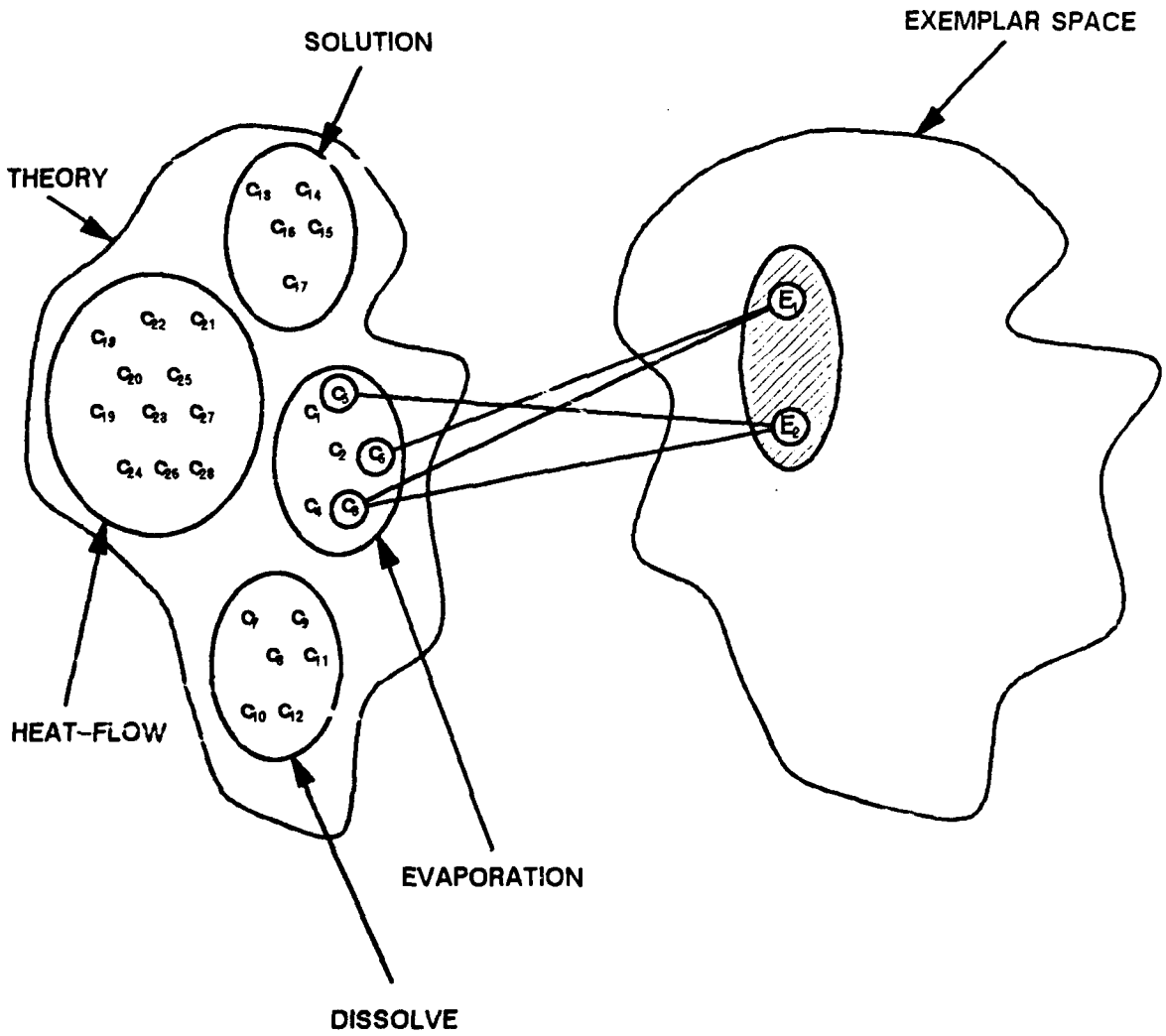
```
Exemplar1:
   Observation: (increase (amount-of vapor))
   Scenario:  <salt-solution-evaporation-scenario>
   Explanation:
      (increase (amount-of vapor))
         I+[(amount-of vapor),
                  (A (evaporation-rate (evaporation salt-solution vapor)))]
            (active (evaporation salt-solution vapor))
               (open? (container salt-solution))
   Components:
      Evaporation:
         I+[(amount-of ?vapor), (A (evaporation-rate ?self))]
         (open? (container ?liquid))

Exemplar2:
   Observation: (decrease (amount-of salt-solution))
   Scenario:  <salt-solution-evaporation-scenario>
   Explanation:
      (decrease (amount-of salt-solution))
         I-[(amount-of salt-solution),
                  (A (evaporation-rate (evaporation salt-solution vapor)))]
            (active (evaporation salt-solution vapor))
               (open? (container salt-solution))
   Components:
      Evaporation:
         I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
         (open? (container ?liquid))
```

Figure 6.7 *The two exemplars formed after the observations* from the scenario in figure 6.5 are explained by the theory.

6.3.2.3. Dissolving Salt in a Salt Solution

Figure 6.9 shows a second scenario that is given to the system. In this scenario, some salt is placed in contact with a solution of salt in water in a closed container. The salt is soluble in the salt solution. In addition, there is a heat-connected path between the salt and the salt solution. However, the temperature of salt is equal to the temperature of the salt solution.

Figure 6.10 shows the behavior of the scenario that is predicted based on the theory of figure 6.4. The salt dissolves in the salt solution. The predicted changes are a decrease in the amount of salt, an increase in the amount of the solute in the salt solution and an increase in the concentration of the salt solution. There are three inactive processes – evaporation of the salt solution which fails because the container is not open and two heat flows – one from the salt solution to the salt and one in the reverse direction – both of which fail because the temperatures of the salt and the salt solution are equal. Hence, the amounts of the vapor and the salt solution and the temperatures of the salt and the salt solution are predicted to remain constant.



E1 = Exemplar for the observed increase in the amount of vapor
E2 = Exemplar for the observed decrease in the amount of the salt solution
c3 = (open? (container ?liquid))
c5 = I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
c6 = I+[(amount-of ?vapor), (A (evaporation-rate ?self))]

Figure 6.8 The theory space and the exemplar space after exemplars for the observed changes in the scenario on figure 6.5 have been constructed.

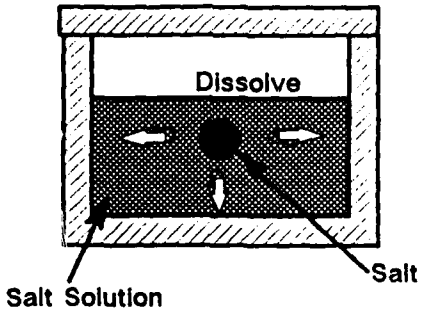


Figure 6.9 A scenario in which salt and a solution of salt in water are placed in contact with each other in a closed container.

```
Behavior2:
   Theory: <Evaporation> <Heat-Flow> <Dissolve> <Solution>
   Scenario: <salt-dissolve-scenario>
   Active Processes:
            (Dissolve salt-solution salt) (solution salt-solution)
   Inactive Processes:
            (Evaporation salt-solution vapor)
            (heat-flow salt salt-solution contact1-path)
            (heat-flow salt-solution salt contact2-path)
   Predicted Changes:
            Increase (amount-of (solute-of salt-solution))
            Decrease (amount-of salt)
            Increase (concentration salt-solution)
   Explanations:
      (Decrease (amount-of salt))
         I-[(amount-of salt), (A (Dissolve-rate (Dissolve salt-solution salt)))]
            Active (Dissolve salt-solution salt)
               (Dissolves? salt salt-solution)
      (Increase (amount-of (solute-of salt-solution)))
         I+[(amount-of (solute-of salt-solution)),
                  (A (Dissolve-rate (Dissolve salt-solution salt)))]
            Active (Dissolve salt-solution salt)
               (Dissolves? salt salt-solution)
      (Increase (concentration salt-solution))
         (Q+ (concentration salt-solution) (amount-of (solute-of salt-solution)))
            (Active (solution salt-solution))
               (Greater-than (A (amount-of (solute-of salt-solution))) 0)
         (Increase (amount-of (solute-of salt-solution)))
            I+[(amount-of (solute-of salt-solution)),
                     (A (Dissolve-rate (Dissolve salt-solution salt)))]
               Active (Dissolve salt-solution salt)
                  (Dissolves? salt salt-solution)
```

Figure 6.10 The predicted behavior for the scenario shown in figure 6.9.

The observations made in the scenario are that the amount of the salt solution remains constant, the concentration of the salt solution is increasing and the amount of salt is decreasing. Figure 6.11 shows the three exemplars resulting from these observations. The first exemplar is based on the observed increase in the concentration of the salt solution. It exemplifies four components of the theory – a quantity condition and a relation of the solution definition and a precondition and an influence of the dissolve process definition. The second exemplar is based on the observed decrease in the amount of salt and exemplifies two components of the theory – a precondition and an influence of the dissolve process definition. The third exemplar is based on the observation that the amount of the salt solution remains constant. It exemplifies two components of the theory – a precondition and an influence of the evaporation process definition. The necessity of the precondition *open?* is illustrated by the exemplar. Figure 6.12 shows the exemplar space corresponding to the theory after the second example.

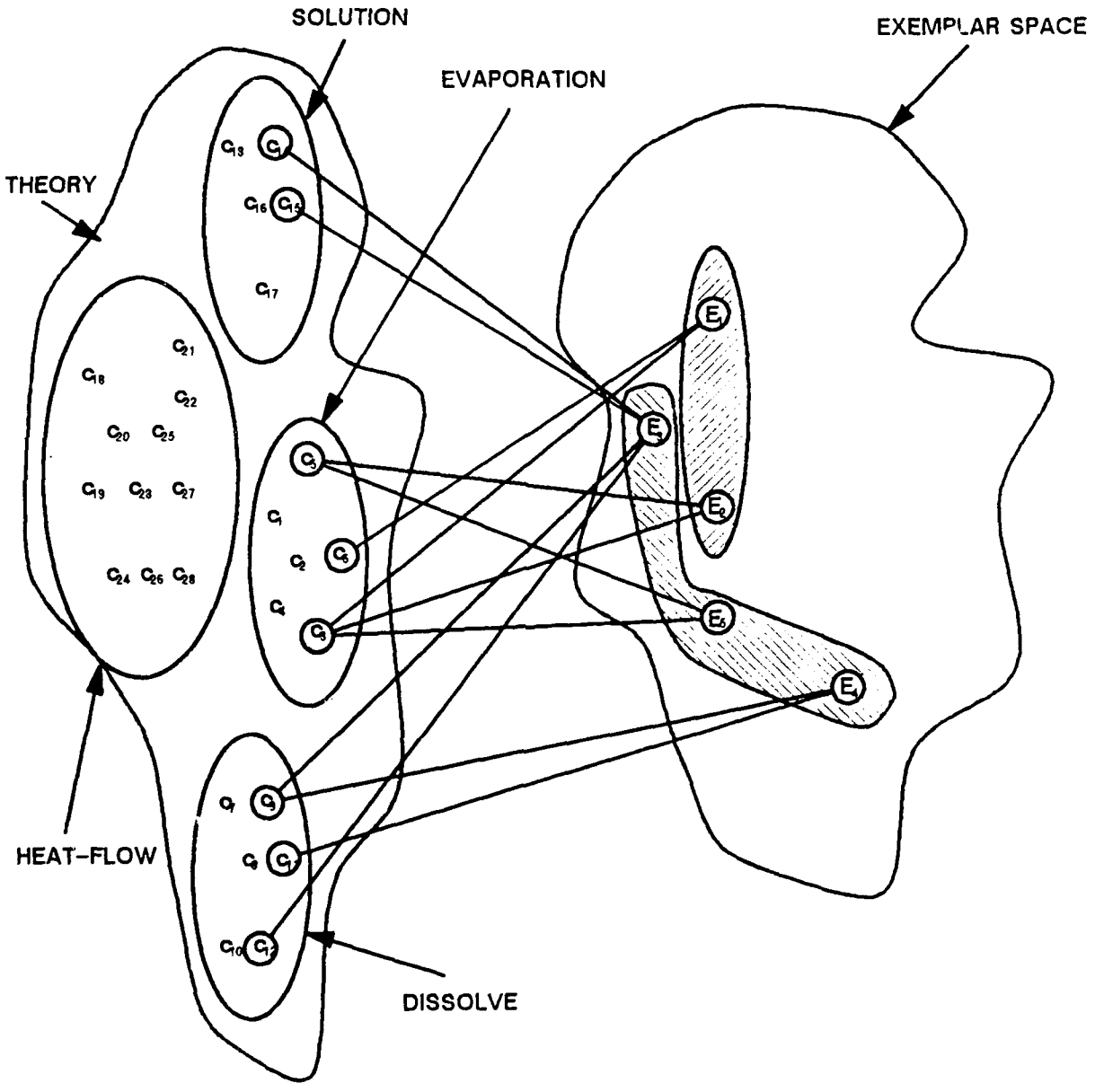
6.3.2.4. Some Additional Examples

Figure 6.13 shows two additional scenarios. In the first scenario, a stone is connected to some ice through the wall of the container. Though the temperature of the stone is much higher than the temperature of the ice, heat flow does not occur because the wall of the container is insulated. This results in two exemplars corresponding to the observations that the temperatures of the stone and the ice remain constant. Both the exemplars illustrate why the *heat-aligned?* precondition is required for the heat flow process.

In the second scenario, a plastic ball is placed in contact with the salt solution in a container. The dissolve process does not occur because the plastic ball is not soluble in the salt solution. Therefore the amount of the plastic ball is observed to remain constant. The exemplar for this observation illustrates why the precondition *dissolves?* is required for the dissolve process. Figure 6.14 shows the relationship between the theory and the exemplar space after these two additional examples.

```
Exemplar3:
   Observation:          (increase (concentration salt-solution))
   Scenario:             <salt-dissolve-scenario>
   Explanation:
      (increase (concentration salt-solution))
         (Q+ (concentration salt-solution) (amount-of (solute-of salt-solution)))
            (Active (solution salt-solution))
               (Greater-than (A (amount-of (solute-of salt-solution))) 0)
         (increase (amount-of (solute-of salt-solution)))
            I+[(amount-of (solute-of salt-solution)),
                  (A (Dissolve-rate (Dissolve salt-solution salt)))]
               Active (Dissolve salt-solution salt)
                  (Dissolves? salt salt-solution)
   Components:
      Dissolve:
         (dissolves? ?solid ?solution)
         I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]
      Solution:
         (greater-than (A (amount-of (solute-of ?solution))) 0)
         (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))

Exemplar4:
   Observation:          (decrease (amount-of salt))
   Scenario:             <salt-dissolve-scenario>
   Explanation:
      (decrease (amount-of salt))
         I-[(amount-of salt), (A (Dissolve-rate (Dissolve salt-solution salt)))]
            Active (Dissolve salt-solution salt)
               (Dissolves? salt salt-solution)
   Components:
      Dissolve:
         (dissolves? ?solid ?solution)
         I-[(amount-of ?solid), (A (dissolve-rate ?self))]

Exemplar5:
   Observation:          (constant (amount-of salt-solution))
   Scenario:             <salt-dissolve-scenario>
   Explanation:
      (constant (amount-of salt-solution))
         I-[(amount-of salt-solution),
                  (A (evaporation-rate (evaporation salt-solution vapor)))]
         (Inactive (evaporation salt-solution vapor))
            (:not (open? (container salt-solution)))
   Components:
      Evaporation:
         I-[(amount-of ?liquid), (A (evaporation-rate (evaporation ?liquid ?vapor)))]
         (open? (container ?liquid))
```

Figure 6.11 The exemplars for the observed changes in the scenario shown in figure 6.9.



E3 = Exemplar for the observed increase in the concentration of the salt solution
E4 = Exemplar for the observed decrease in the amount of salt
E5 = Exemplar for the observation that the amount of the salt solution is constant
c3 = (open? (container ?liquid))
c5 = I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
c9 = (dissolves? ?solid ?solution)
c11 = I-[(amount-of ?solid), (A (dissolve-rate ?self))]
c12 = I+[(amount-of (solute-of ?solution)), (A (dissolve-rate ?self))]
c14 = (greater-than (A (amount-of (solute-of ?solution))) 0)
c15 = (Q+ (concentration ?solution) (amount-of (solute-of ?solution)))

Figure 6.12 The theory space and the exemplar space after exemplars for the observations in the scenario of figure 6.9 have been added.

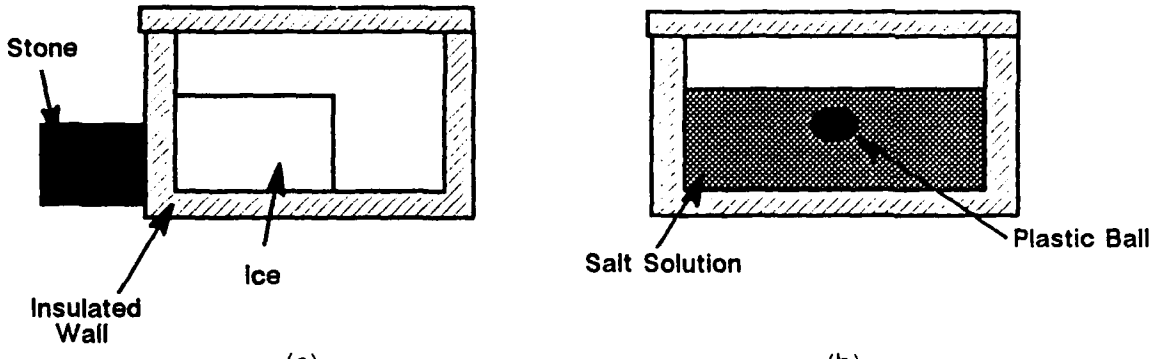


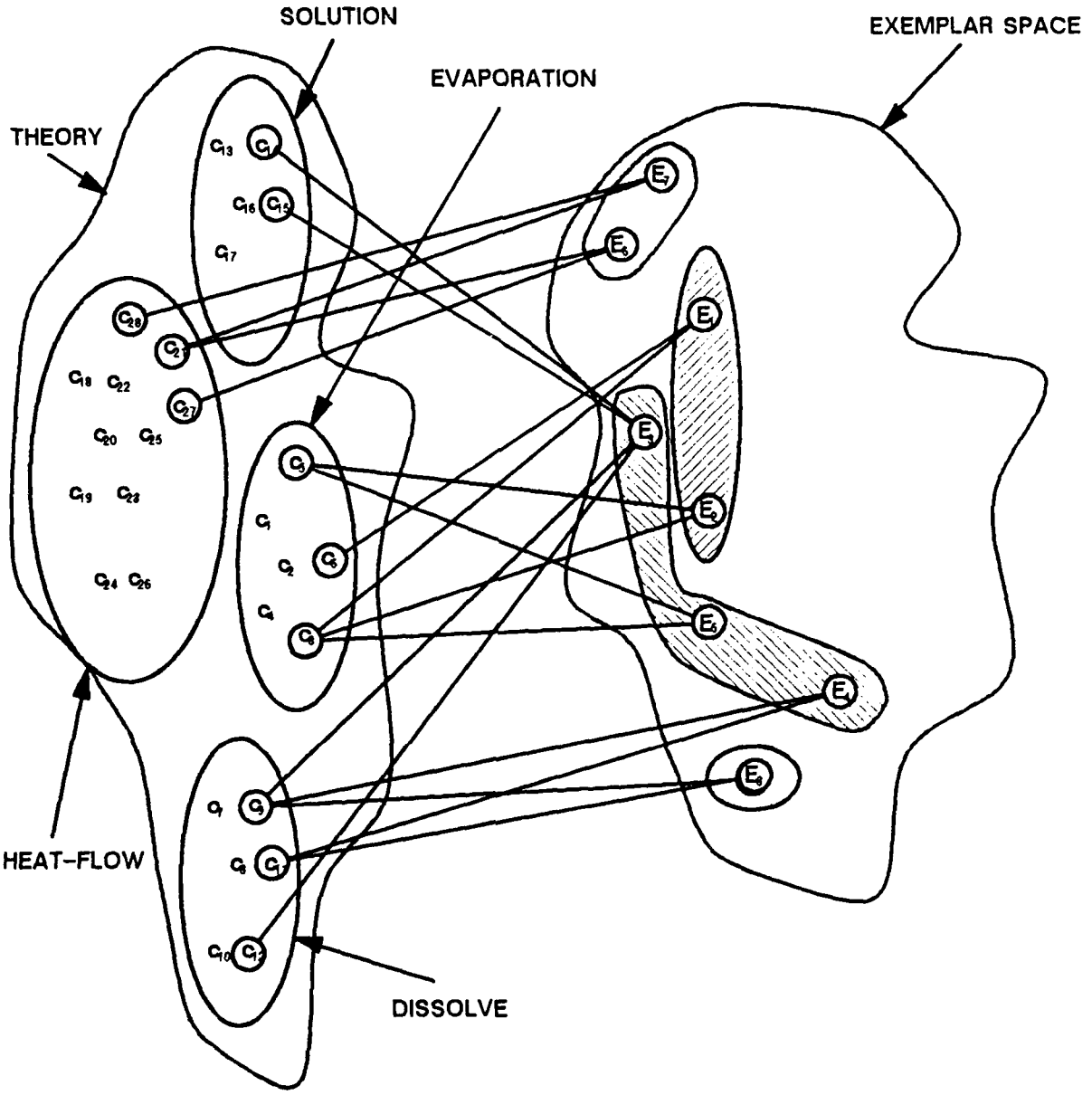
Figure 6.13 (a) A scenario in which ice is placed in a container. The wall of the container is insulated against heat flow. A stone which is at a temperature lower than that of ice is in contact with the container's wall. (b) A scenario in which a plastic ball is placed ina salt solution in a closed container.

6.3.3. Using the Exemplar Space

The exemplar space can be used to test revised theories. For each revised theory, the procedure for testing the theory involves:

Exemplar Retrieval:

The exemplars whose observations have to be re-explained because the revision may have invalidated the earlier explanations are retrieved. Each component of the theory has associated with it a set of exemplars that describe how the component is used. In the case of components that are effects of a process, the exemplars are those which use the component to construct an explanation for the exemplar observation. In the case of components that are conditions of a process, there are two types of exemplars: 1) Exemplars that describe the necessity of the condition. In these exemplars, the condition is not satisfied in the given scenario and as a result the process is inactive. The failed condition is used in the construction of an explanation for why the process is not active. 2) Exemplars which illustrate the active process, that is, all the conditions are satisfied. The component is used in the construction of an explanation for why the process is active. As is described in section 6.3.1, the exemplars that are retrieved depends on the type of revision and the component revised. There are three types of retrieval: 1) Retrieving all the exemplars in which a process has been active. This corresponds to fetching all the



E6 = Exemplar for the observation that the temperature of the stone is constant
E7 = Exemplar for the observation that the temperature of the ice block is constant
E8 = Exemplar for the observation that the amount of the plastic ball is constant
c9 = (dissolves? ?solid ?solution)
c11 = I-[(amount-of ?solid), (A (dissolve-rate ?self))]
c21 = (heat-aligned? ?path)
c27 = I-[(temperature ?source), (A (heat-flow-rate ?self))]
c28 = I+[(temperature ?destination), (A (heat-flow-rate ?self))]

Figure 6.14 The theory space and the exemplar space after exemplars for the observations in the scenarios of figure 6.13 have been added.

exemplars associated with the effects of the process. 2) Retrieving all the exemplars in which a process is not active due to failed conditions. This corresponds to fetching the exemplars associated with the failed condition in which the process is not active. 3) Retrieving all the exemplars which use a specified component to construct an explanation. This corresponds to fetching all the associated exemplars for the component.

Explanation Reconstruction:

The revised theory is used to explain the exemplar observation for the exemplar scenario. If the revised theory cannot explain the observation then it is rejected.

Formation of Exemplar Spaces:

If the revised theory can re-explain the observations of all the retrieved exemplars, then it is consistent with the past observations of the system as represented by the exemplar space. An exemplar space corresponding to this theory is created. Since a number of revised theories may be consistent with the past observations, distinct exemplar spaces have to be created for each theory. Then, if more than one theory is used to explain future observations, the exemplar space of each theory can be augmented separately. Also, exemplar-based theory rejection can be used on each theory separately. The exemplar space corresponding to the revised theory consists of the union of the exemplars that are not retrieved in step 1 (and, hence were not affected by the revisions) and the exemplars that are constructed based on the explanations generated in step 2.

Since there is no relation between the explanation based on the original theory and the explanation based on the revised theory, some components of the revised theory can lose exemplars (if the new explanation does not need the components) or gain exemplars (if the new explanation uses the components while the old does not). Consider a component C which has two exemplars, E1 and E2. Suppose, the revised theory is such that E1 is retrieved in step 1. Also, suppose that C is not revised to generate the revised theory. The re-explanation for the observed change may not use the component C. In this case, in the revised theory, C will not have E2 as an exemplar. This is the reason why more than one exemplar of each type is retained for each component. C can lose all its exemplars if each exemplar is retrieved and the re-explanations do not use C. This causes a problem because if C is revised later then there will be no exemplars associated with C to test the revision. If a

large number of components have no exemplars associated with them then the usefulness of exemplar–based theory rejection decreases. This problem can be alleviated by ensuring that the threshold in the exemplar threshold criterion is high.

6.3.3.1. Examples

Examples involving revisions to the theory described earlier are used to illustrate the use of exemplar spaces to test revised theories.

6.3.3.2. Evaporation of Alcohol

Figure 6.15 describes a scenario in which alcohol is placed in an open container in contact with its

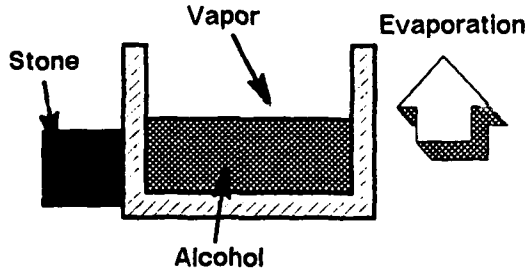


Figure 6.15 A scenario in which alcohol is placed in an open container in contact with its vapor. A stone is touching the wall of the container. The wall is insulated against heat flow.

vapor. The container is in contact with a stone which is at a much lower temperature than alcohol. There are two heat paths – one between the alcohol and the vapor through the surface and the other between the alcohol and the stone through the container. However, both these paths are insulated against heat[1]. Section 3.5.1 describes the behavior predicted by a similar theory for a similar scenario. The temperature of the alcohol is observed to decrease and this change is not predicted by the theory. This results in an unexpected observation. Chapter 4 describes the revisions that can be made to the evaporation and heat flow process definitions to generate theories that can explain the observed decrease in the temperature of alcohol. Three such revisions are:

1) Adding a New Influence:

A new influence is added to the evaporation process:

[1] Recall that the vapor is treated as a unit and though the temperature in layers very close to the liquid will change, the temperature of the whole vapor remains constant.

→ I-[(temperature ?liquid), (A (evaporation-rate ?self))].

This revision results in the explanation shown in figure 6.16 for the observed decrease in the temperature of alcohol.

```
(decrease (temperature alcohol))
  I-[(temperature alcohol), (A (evaporation-rate (evaporation alcohol vapor)))]
    (Active (evaporation alcohol vapor))
      (open? (container alcohol))
    H: New-Influence? I-[(temperature alcohol),
                         (A (evaporation-rate (evaporation alcohol vapor)))]
```

Figure 6.16 The explanation for the observed decrease in the temperature of alcohol based on the revised definition for evaporation.

2) Deleting a Precondition:

The failed precondition *heat-aligned?* of the heat flow process is deleted from the process definition:

(heat-aligned? ?path) →.

This results in the explanation shown in figure 6.17.

```
(decrease (temperature alcohol))
  I-[(temperature alcohol), (A (heat-flow-rate (heat-flow alcohol stone container-path)))]
    (Active (heat-flow alcohol stone container-path))
      H: Delete Condition (heat-aligned? container-path)
      (greater-than (A (temperature alcohol)) (A (temperature stone)))
```

Figure 6.17 The explanation for the observed decrease in the temperature of alcohol based on the revised heat flow process.

3) Widening the Scope:

The scope of the precondition *heat-aligned?* of the heat flow process definition is widened so that it is satisfied in the given scenario.

(heat-aligned? ?path) → (aligned? ?path).

This results in the explanation shown in figure 6.18.

(decrease (temperature alcohol))
I-[(temperature alcohol), (A (heat-flow-rate (heat-flow alcohol stone container-path)))]
(Active (heat-flow alcohol stone container-path))
H: Widen-Scope (heat-aligned? container-path)
(aligned? container-path)
(greater-than (A (temperature alcohol)) (A (temperature stone)))

Figure 6.18 The explanation for the observed decrease in the temperature of alcohol based on the revised heat flow process definition.

The exemplar space constructed in the previous section is used to test these three revised theories.

New Influence:

The exemplars retrieved due to the added influence are the exemplars for the effects of the evaporation process. These correspond to the exemplars E_1, E_2 and E_6 of figure 6.14. The revised theory can also explain the exemplar observations – the decrease in the amount of the salt solution and the increase in the amount of the vapor in the exemplar scenario of figure 6.5 and the amount of the salt solution remaining constant in the exemplar scenario of figure 6.9. In fact, the added influence is such that the explanations are not affected. Therefore, the revised theory is consistent with the exemplars. The exemplar space corresponding to this revised theory has a new exemplar for the observed decrease in the temperature of alcohol.

Delete Precondition:

The exemplars retrieved for the deleted precondition are those exemplars of the failed condition *heat-aligned?* that are used to construct explanations for why the heat flow process is inactive due to a failure of this condition. There are two such exemplars in figure 6.14 – E_6 and E_7. The exemplar observations are that the temperatures of the stone and the ice block are constant in the scenario shown in figure 6.13a. However, the revised theory predicts that the heat flow process from the stone to the ice is active in the exemplar scenario and therefore, the temperatures of the two objects are changing. This is not consistent with the exemplar observations. Hence, the revised theory is rejected.

Widen Scope:

Since the condition is less constrained by the revision, the exemplars retrieved are those exemplars of the failed condition *heat-aligned?* which are used to construct explanations for

why the heat flow process is inactive. In this case, the exemplars E_6 and E_7 of figure 6.14 are retrieved. The exemplar observations are that the temperatures of stone and the ice block are constant in the scenario shown in figure 6.13a. The revised theory again predicts that the heat flow process is active and that the temperatures of the stone and the ice are changing. Since this is not consistent with the exemplar observations the revised theory is rejected.

Only one revised theory (corresponding to the new influence for evaporation) successfully meets the exemplar consistency test. The exemplar space corresponding to the revised theory is shown in figure 6.19. The exemplar space has a new exemplar that illustrates the use of the new influence component that is added to the evaporation process definition.

6.3.3.3. Evaporation of a Sugar Solution

As another example, consider the scenario shown in figure 6.20. A solution of sugar is placed in an open container. A wooden ball is placed in contact with the sugar solution in the container. The scenario includes facts such as the container is open and the wooden ball does not dissolve in the sugar solution. According to the theory described in figure 6.4, there is one active process – evaporation of the sugar solution, and one inactive process – dissolving of the wooden ball in the sugar solution – which fails because it is not soluble. Therefore, the theory predicts that the amount of the solution decreases and the amount of the vapor increases. However, the concentration of the sugar solution is also observed to increase resulting in an unexpected observation.

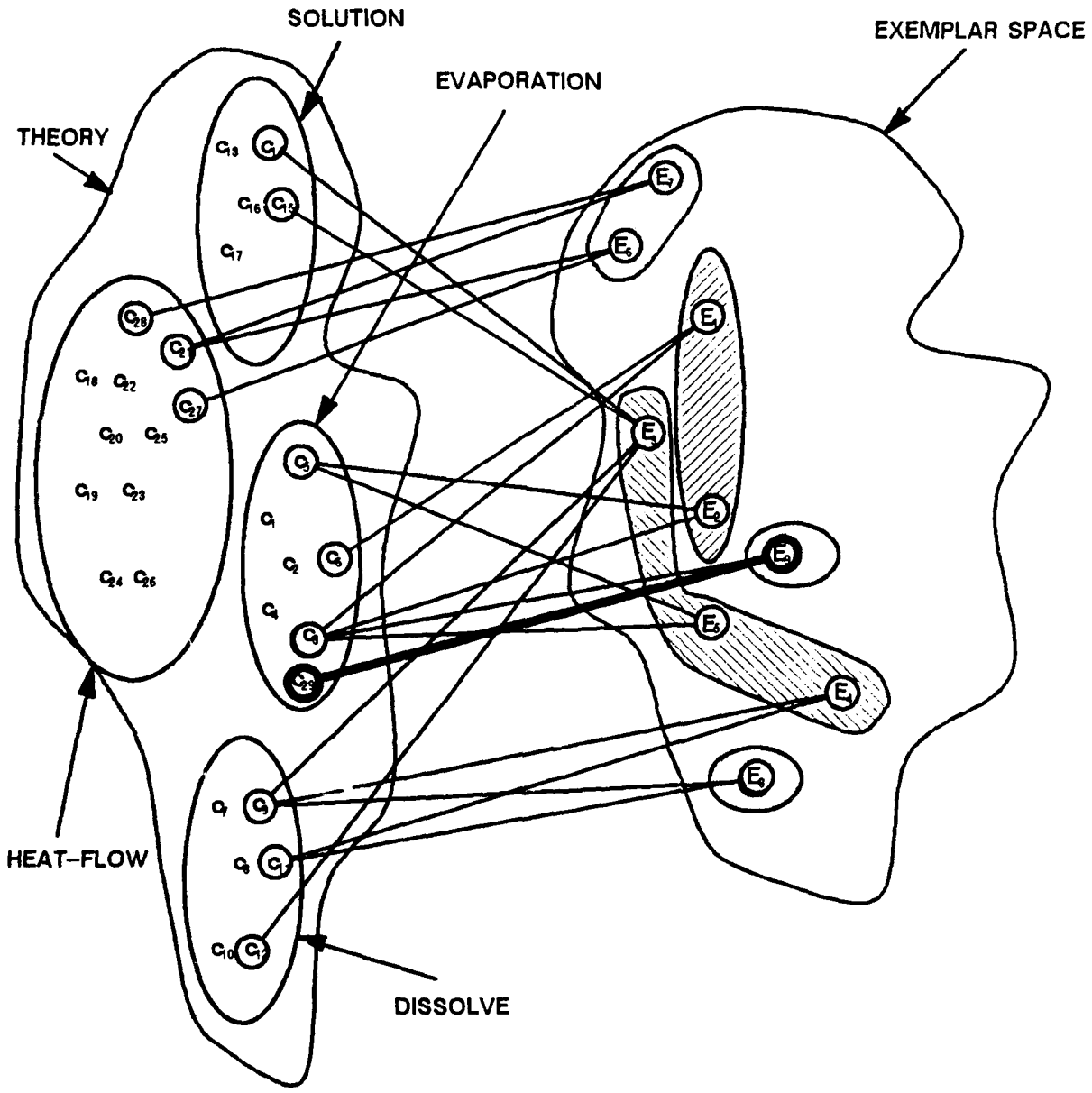
Two of the revisions that can be made to the theory which explain the observed increase in the concentration of the sugar solution are:

1) Narrowing the Scope:

The scope of an influence of evaporation is narrowed so that it affects only the solvent of the solution if the participating liquid is a solution:

```
I-[(amount-of ?liquid), (A (evaporation ?self))] →
        I-[(amount-of (solvent-of ?liquid)), (A (evaporation-rate ?self))].
```

The explanation for the observed increase in the concentration of the sugar solution based on the revised theory is shown in figure 6.21.



E9 = Exemplar for the observed decrease in the temperature of alcohol

C3 = (open? (container ?liquid))

c29 = I-[(temperature ?liquid), (A (evaporation-rate ?self))]

Figure 6.19 The revised theory space (that includes the new influence for evaporation) and the exemplar space corresponding to the revised theory. It has an exemplar based on the failure observation – the unexpected decrease in the temperature of alcohol.

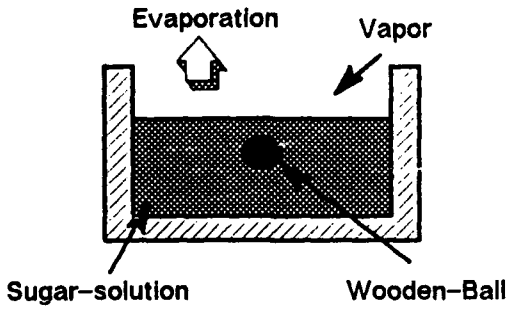


Figure 6.20 The scenario for the second example. A solution of sugar in water is placed in an open container. A wooden ball is also placed in the container in contact with the sugar solution.

```
(Increase (concentration sugar–solution))
   (Q– (concentration sugar–solution) (amount–of (solvent–of sugar–solution)))
      (Active (solution sugar–solution))
         (Greater–than (A (amount–of (solute–of sugar–solution))) 0)
   (Decrease (amount–of (solvent–of sugar–solution)))
      I–[(amount–of (solvent–of sugar–solution)),
                  (A (evaporation–rate (evaporation sugar–solution vapor)))]
         Active (evaporation sugar–solution vapor)
            (open? (container sugar–solution))
      H: Narrow Scope I–[(amount–of ?liquid), (A (evaporation ?self))]
```

Figure 6.21 The explanation for the observed increase in the concentration of the sugar solution based on the revised influence.

2) Deleting a Precondition:

The failed precondition *dissolves?* of the dissolve process is deleted.

(dissolves? ?solid ?solution) →.

The explanation for the observed increase in the concentration of the sugar solution based on this revision is shown in figure 6.22.

```
(Increase (concentration sugar-solution))
   (Q+ (concentration sugar-solution) (amount-of (solute-of sugar-solution)))
      (Active (solution sugar-solution))
         (Greater-than (A (amount-of (solute-of sugar-solution))) 0)
   (Increase (amount-of (solute-of sugar-solution)))
      I+[(amount-of (solute-of sugar-solution)),
                  (A (dissolve-rate (dissolve wooden-ball sugar-solution)))]
         Active (dissolve wooden-ball sugar-solution)
            H: Delete Condition (dissolves? ?solid ?solution)
```

Figure 6.22 The explanation for the observed increase in the concentration of the sugar solution based on the revised definition for dissolve.

The exemplar space constructed in figure 6.14 can be used to test these two revised theories:

Narrow Scope:

The exemplars retrieved for this revision are the exemplars associated with the influence component, that is, exemplars E_2 and E_5 in figure 6.14. The revised theory also provides explanations for the exemplar observations. The explanation for the observed decrease in the amount of salt-solution for the exemplar E_2 is shown in figure 6.23.

```
(decrease (amount-of salt-solution))
   (decrease (amount-of (solvent-of salt-solution)))
      I-[(amount-of (solvent-of salt-solution)),
                        (A (evaporation-rate (evaporation salt-solution vapor)))]
         (active (evaporation salt-solution vapor))
            (open? (container salt-solution))
   (Q+ (amount-of salt-solution) (amount-of (solvent-of salt-solution)))
      (active (solution salt-solution))
         (greater-than (a (amount-of (solute-of salt-solution))) 0)
```

Figure 6.23 The explanation for the observed decrease in the amount of the salt solution based on the revised theory.

Delete Precondition:

The exemplars retrieved for this revision are the exemplars of the dissolve process being inactive due to the precondition *dissolves?* having failed. This corresponds to the exemplar E_8 of figure 6.14. The revised theory predicts that the amount of the plastic ball will decrease because the dissolve process is active in the scenario of figure 6.13b. Since the amount of the plastic ball is observed to remain constant this prediction is inconsistent with the exemplar. Therefore, the revised theory is rejected.

Figure 6.24 shows the exemplar space corresponding to the revised theory that incorporates the modified influence of the evaporation process. The exemplars E_2 and E_5 have been modified because new explanations have been constructed using the revised theory.

6.4. Evaluation of Exemplar-based Theory Rejection

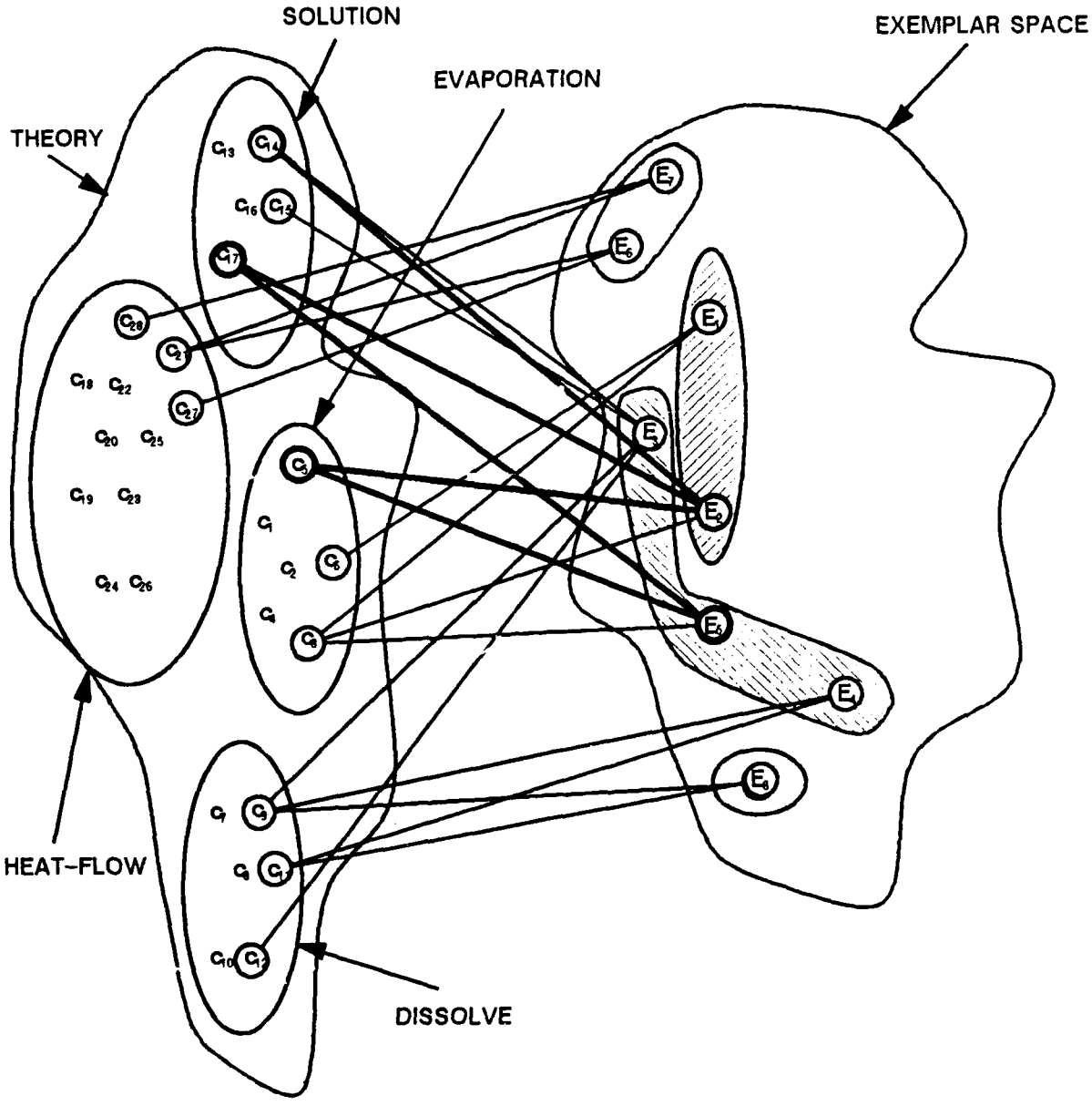
The performance of exemplar-based theory rejection is primarily governed by two functions: the *retention function* which determines if a newly created exemplar is stored in the exemplar space and the *retrieval function* which determines the exemplars that are retrieved to test a revised theory. The retention function depends on a pre-specified threshold limit on the number of exemplars for each component and the criteria for determining the prototypicality of an exemplar. The retrieval function depends on the types of the components revised and the types of revisions performed to obtain the revised theory. Figure 6.25 shows four different types of retrieval functions. An *exact* retrieval function retrieves exactly those exemplars that are affected by the revisions and which require re-explanation. An *over-estimating* retrieval function fetches exemplars that are not affected by the revision (the new explanation is identical to the existing explanation) in addition to all the exemplars that are affected. An extreme form of an over-estimating function is a function that retrieves all the exemplars in the exemplar space. An *under-estimating* retrieval function fails to retrieve all the exemplars that are affected by the revision.

To recapitulate, the retention function used by COAST is:

Retention Function (exemplar):
Exemplar is retained only if at least one of the conditions specified below is met:

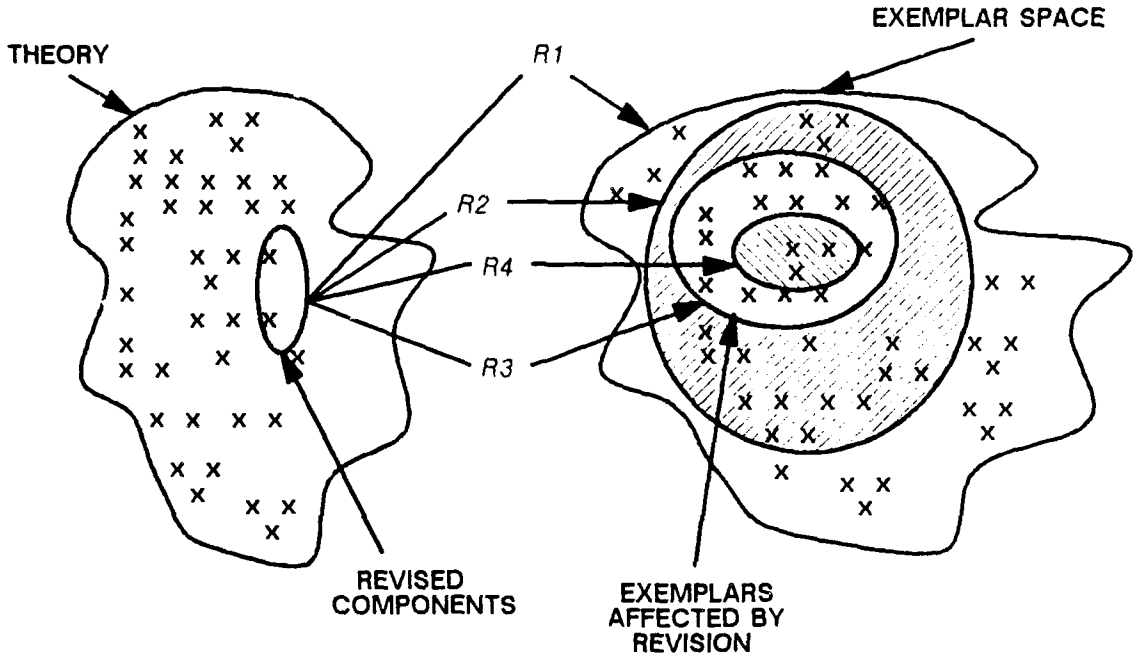
1) The number of exemplars of any of the components illustrated by the exemplar is less than the threshold limit.
2) The exemplar is a prototype for one of the components illustrated by the exemplar (prototypicality is defined based on the simplicity of the exemplar scenario and the exemplar explanation).

The retrieval function used by COAST is:



E2 = Exemplar for the observed decrease in the amount of brine
E5 = Exemplar for the observation that the amount of brine remains constant
c5 = I-[(amount-of (solvent-of ?liquid)), (A (evaporation-rate ?self))]
c14 = (greater-than (A (amount-of (solute-of ?solution))) 0)
c17 = (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution)))

Figure 6.24 The revised theory space (with the scope of the influence of evaporation narrowed) and the changed exemplar space.



R1 Retrieval function that retrieves all the exemplars in the exemplar space.

R2 An over-estimating retrieval function.

R3 An exact retrieval function.

R4 An under-estimating retrieval function.

Figure 6.25 Different retrieval functions.

Retrieval Function (revised-theory):
If the revised component is:

1) A condition of a process and the revision loosens the constraints on the process then the exemplars of the process being inactive due to the failure of the revised condition are retrieved.
2) A condition of a process and the revision further constrains the process then all the exemplars in which the process is active are retrieved.
3) An effect of a process and the revision limits the effects of the process then the exemplars of the revised effect are retrieved.
4) An effect of a process and the revision augments the effects of the process then all the exemplars in which the process is active are retrieved.

If there is more than one revised component then the union of all the exemplars retrieved for each revised component is returned.

The retrieval function used by COAST is an over-estimating retrieval function since, in general, exemplars that are not affected by the revision can be retrieved for revised theories that include the

revision of an effect that augments the previous effects of the process. Also, the retrieval function for COAST is constructed in such a manner that it always retrieves all the exemplars in the exemplar space that are affected by the revisions, that is, it never under-estimates.

The following subsections analyze three characteristics of exemplar-based theory rejection and show how the implemented functions in COAST can be improved.

6.4.1. Efficacy

The *efficacy* of an exemplar-based theory rejection system is defined to be the ability of the method to eliminate theories that cannot explain previously encountered observations. Both the retention function and the retrieval function, play an important role in determining whether the method is efficacious. If the retention function fails to retain relevant observations or if the retrieval function fails to retrieve stored exemplars of observations that cannot be explained by the proposed theory then the method is less efficacious. The efficacy of exemplar-based theory rejection can be improved by:

1) Increasing the threshold limit to reduce the possibility of losing relevant observations. In practice, this is not a desirable solution since it results in large, unmanageable exemplar spaces.

2) Improving the definition of prototypicality. COAST currently defines prototypicality in terms of the *simplicity* of the scenario and the explanation. If prototypicality can be defined based on how well an exemplar *typifies* the use of the components of the theory then the exemplar observations that are more relevant for testing revised theories can be retained. However, the notion of typicality is harder to define and compute. This is a topic of future research.

3) Constructing an exact or over-estimating retrieval function. The retrieval function used by COAST is an over-estimating function. If an under-estimating retrieval function is used then some of the affected exemplars that could potentially eliminate the revised theory are not retrieved. In addition, the use of an under-estimating retrieval function undermines the integrity of the new exemplar spaces that are created for the successfully tested revised theories since the explanations for the unretrieved affected exemplars are different. Consequently, such exemplars may not include the correct set of components for the revised theory. This has deleterious effects on future retrieval.

6.4.2. Efficiency

The efficiency of an exemplar-based theory rejection system is defined as the ratio of the number of exemplars retrieved to test a revised theory to the number of exemplars actually affected by the revisions. The efficiency of the method is improved by employing a retrieval function that approximates the exact retrieval function as closely as possible. The retrieval function used by COAST can be made to more tightly approximate the exact retrieval function by further breaking up the various types of revisions that are feasible for each type of component of the theory and by equipping the retrieval function with the ability to compute the ramifications of the revisions. For example, consider a revised theory constructed by adding a new influence to a process. The retrieval function currently used by COAST fetches all the exemplars in which the process is active. However, consider an exempla which is retrieved because it uses other effects of the process to explain the observation. If none of the quantities involved in the explanation are affected by the new influence then the explanation itself is not affected. Such an exemplar does not contribute to the testing of the revised theory. In order for the retrieval function to identify and prevent the retrieval of such exemplars it must be able to trace through the active influences and qualitative proportionalities to determine the scope of the newly added influence.

An additional factor that determines the efficiency of an exemplar-based theory rejection system, and which has been ignored in above definition, is the cost of re-explaining the retrieved exemplars using the revised theory. Related issues such as the effect of the size of the exemplar space on the efficiency of the method and the trade-offs involved in using exemplar spaces composed of a small number of complex exemplars versus a large number of simple exemplars are a topic of future investigation and have not been addressed by the current research.

6.4.3. Oscillation

A theory revision system is defined to *oscillate* if, starting from an initial theory, a series of revisions eventually results in a theory that is functionally identical to the initial theory or one of the intermediate theories. A theory revision system that uses exemplar-based theory rejection to test revised theories can be prevented from oscillating if the exemplar-based theory rejection system satisfies two conditions: 1) the retention function stores every observation that invoked theory revision as an exemplar, and 2) the retrieval function is exact or over-estimating. The proof is fairly obvious. Since the revised theory is tested by exemplar-based theory rejection, it can explain all

the previous observations that resulted in theory revision (the exemplar for each failure observation is stored by the augmented retention function and is retrieved, if necessary, by the over-estimating or exact retrieval function). Therefore, it cannot be identical to any of the previous theories because each of the theories could not explain at least one of the failure observations (and therefore had to be revised). COAST's retention function can be augmented to include the above condition. In this case, COAST will not oscillate.

In practice, the first condition, requiring every observation that led to the failure of a theory be stored as an exemplar, is not difficult to satisfy. First, the exemplar for the failure observation is retained by the revised theory as an illustration of the use of the revised components unless both the retention conditions are not met. For example, the exemplar may not be retained in some cases if the revisions involve deletion of components. The retention function must therefore be augmented to include a condition specifying that every exemplar that invoked theory revision must be retained in the exemplar space. Second, a failure to explain an observation using the theory is an abnormal occurrence compared to the successful explanation of observations. Consequently, the exemplars due to failure observations usually form a small fraction of the total exemplars in the exemplar space.

6.5. Discussion

This chapter has described a method for testing revised theories based on exemplars. PROTOS [Bareiss87] also uses exemplars and prototypes. However, there are considerable differences in the representation, indexing and use of exemplars. Exemplar-based theory rejection includes an explanation for how the components of the theory are used. Also, the retrieval of exemplars in PROTOS is similarity-based, whereas, in exemplar-based theory rejection, exemplars are retrieved based on the revised components of the theory. Exemplar-based theory rejection uses exemplars to preserve consistency with past observations whereas PROTOS uses exemplars for classifying new cases. There are also similarities between exemplar-based theory rejection and other approaches based on past examples such as case-based reasoning [Hammond86, Kolodner87], the use of positive examples in similarity-based learning (also, termed empirical learning) [Dietterich83, Mitchell78], incremental and nonincremental similarity-based learning [Michalski83a, Quinlan83] and the use of precedents and history in analogical learning [Falkenhainer87a, Winston83]. Case-based reasoning involves the access of previous *similar* cases; exemplar-based theory rejection retrieves only those exemplars that are relevant to the revisions as determined by

explanations. Also, case-based reasoning does not incorporate a notion of maintaining consistency with past experiences. Positive examples are used in similarity-based learning to grow concepts. However, unlike exemplar-based theory rejection, these methods do not incorporate explanations for the positive examples. The use of positive examples in incremental similarity-based learning is analogous. Theories are refined incrementally based on the new observations and the exemplars and an analogy can be drawn between the theory revision process and the concept description refinement process in incremental learning. However, the major distinction is that exemplar-based theory rejection incorporates explanations during the refinement process. The use of precedents and past experiences in analogical learning also differs significantly from exemplar-based theory rejection. These methods store and access precedents using primarily similarity-based techniques.

This chapter has described a method called exemplar-based theory rejection that tests revised theories. The method selectively collects examples that illustrate the use of the theory to explain observations previously accounted for by the system. These examples are organized in an exemplar space that is associated with the components of the theory. The method tests the proposed revisions to a theory by selecting those exemplars from the exemplar space whose explanations are affected by the revisions and verifies that the revised theory can explain the observations in these exemplars. The chapter described how the exemplars are created, how the exemplar spaces are incrementally formed and how the proposed revisions are tested. The method returns those revised theories that are consistent with the observations made previously as represented in the form of exemplars in the exemplar space. In addition, the factors governing the performance of exemplar-based theory rejection systems and the improvements that can be made to COAST's implementation are described.

CHAPTER 7

SELECTION OF THEORIES

7.1. Introduction

Chapters 3 to 6 described how problems with a theory are detected, how revised theories are hypothesized, and how the hypothesized theories are tested and incorrect theories are eliminated. In particular, experimentation-based hypothesis refutation eliminates theories that are not consistent with the experimental observations and exemplar-based theory rejection eliminates theories that are not consistent with the previously established exemplar observations. However, a number of theories may be left even after these two methods have been applied. Some of the remaining theories may be incorrect but were not eliminated due to limitations in the two testing methods. Experiments to eliminate incorrect theories may have failed because the required scenarios could not be constructed. The existing exemplar space may not have examples of scenarios in which the incorrect theories fail to explain an observation. The other remaining theories may be correct but may be notational variants of each other.

Multiple theories pose a problem because the problem solver has to decide which theory to use to analyze future scenarios. It is not practical to retain all the theories to analyze future scenarios due to the inordinate amount of computational resources that will be required. Therefore, it is important to establish a threshold on the number of theories to be retained. Consequently, criteria for selecting theories are required.

This chapter presents three criteria for selecting theories based on aesthetic considerations of the theories. The next section describes the three criteria. The third section describes how COAST computes estimates for each criterion. The last section discusses other work pertaining to the selection of a best theory from competing theories and presents a summary of the chapter.

7.2. Theory Selection Criteria

Explanation–based theory revision uses three criteria for comparing the competing theories that remain after experimentation–based hypothesis refutation and exemplar–based theory rejection:

[a] The Structural Simplicity of the Theory

A theory that is structurally simpler than the other theories is preferred. This criterion is based on the principle of parsimony – of theories of equal explanatory power prefer theories with fewer components. Simpler theories are preferred because they are easier to use.

[b] The Simplicity of the Explanations Constructed by the Theory

A theory that provides simpler explanations for observations is preferred. This criterion is based on the Occam's razor principle which states that, of theories of equal explanatory power, the theory that provides simpler explanations is to be preferred.

[c] The Predictive Power of the Theory

A theory that makes more predictions for scenarios is preferred. This criterion emphasizes a fundamental function of theories – the ability to make predictions. A theory that makes more predictions is preferred because it is easier to determine whether such a theory is incorrect by designing experiments that attempt to falsify the predictions [Popper68]. In addition, if a theory makes many predictions then as these predictions are confirmed by future observations the belief in the theory increases.

7.3. Computing Estimates for the Criteria in COAST

This section discusses various measures for estimating the above criteria for theories represented by Qualitative Process theory.

7.3.1. The Structural Simplicity of the Theory

Different syntactic measures for estimating the structural simplicity of a theory can be defined based on factors such as the number of different types of components in the theory, the number and type of new components introduced by the revisions, preference for certain types of components over other components, etc. Each metric emphasizes different aspects of a theory. A metric that depends on the number of components in a theory specifies a preference for revised theories that are obtained by modifying existing components to revised theories that are obtained

by adding new components to the original theory. Likewise, a metric that establishes a preference for certain types of components over other types of components rates revised theories that involve the addition of the preferred components over those that involve the addition of other components.

COAST employs a metric for estimating the complexity of theories that is based on the number of processes in the theory and the number of components in each process. Figure 7.1 shows the procedure used to compute the metric. The metric results in a preference for revised theories that involve modifications to existing components of a process rather than the addition of new components to a process. Furthermore, the addition of a new process significantly adds to the complexity of a theory as is reflected by the weight of the process and the complexity due to the components introduced by the new process.

```
Procedure Complexity-of-Theory (theory)
   ;;; Weight-of-process and weight-of-component are user-supplied parameters.
   complexity(theory) = 0
   For each process in the theory do
      complexity = complexity + weight-of-process
      For each component in the process do
         ;;; Components of a process are the individuals, preconditions, quantity conditions
         ;;; relations and influences of the process.
         complexity = complexity + weight-of-component
```

Figure 7.1 The procedure for computing the complexity of a theory.

Figure 7.2 shows a process description for the evaporation of liquids. The complexity of a theory consisting of the single process, evaporation, is computed in the figure (the values used for the weights of a process and each of its components are five and one respectively).

Example Illustrating the Comparison of Theories based on Structural Simplicity

Consider an initial theory consisting of process definitions for evaporation, dissolving of a substance, and fluid flow, and an individual view for solutions (figure 7.3). This theory cannot explain an observed increase in the concentration of the salt solution in the scenario shown in figure 7.4. Revised theories that can explain this observation are obtained as described in chapter 4 and are tested using experiments and exemplars as described in chapters 5 and 6. Suppose three revised theories remain after the testing stage:

```
Evaporation (?liquid ?vapor)
   Individuals
      ?liquid ?vapor
   Preconditions
      (open? (container ?liquid))
   Quantity Conditions
   Relations
      (Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor))
   Influences
      I-[(amount-of ?liquid), (A (evaporation-rate ?self))]
      I+[(amount-of ?vapor), (A (evaporation-rate ?self))]

Complexity of the theory = Weight-of-process +
                           Number of components * Weight-of-component
                         = 5 + 6 * 1
                         = 11
```

Figure 7.2 The complexity of the evaporation process.

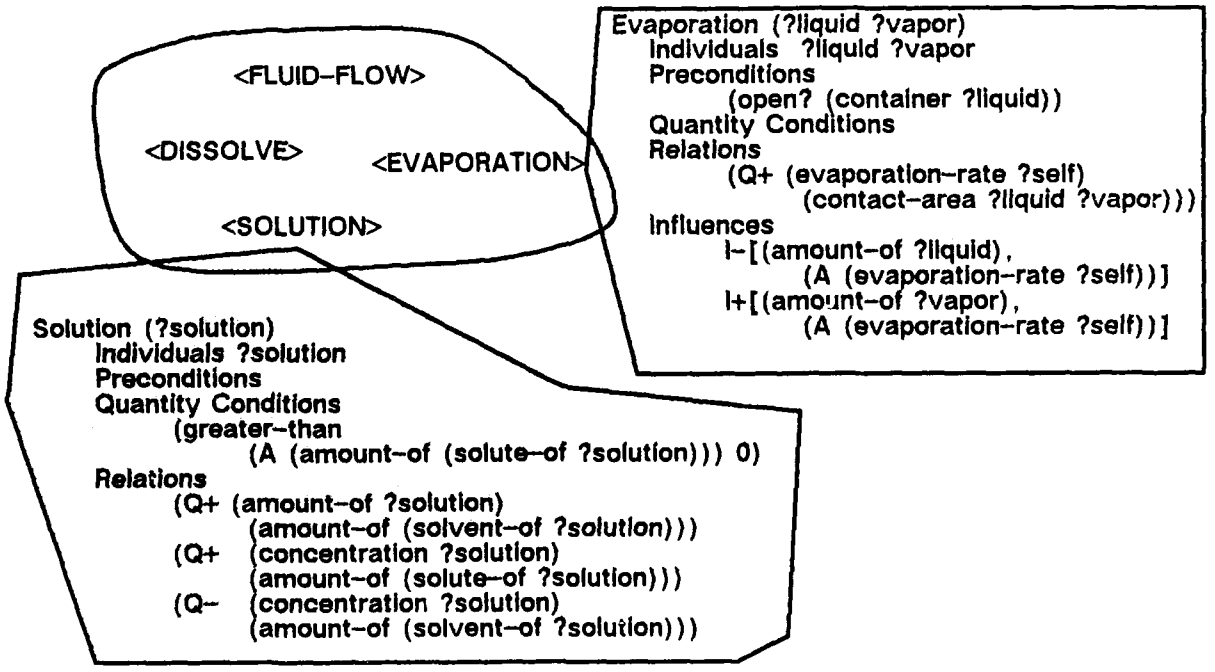


Figure 7.3 A theory describing fluid-flow, evaporation, dissolve and solutions. The detailed descriptions for evaporation and solutions are shown.

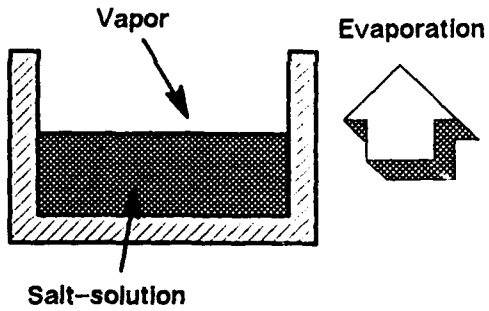


Figure 7.4 A scenario in which a salt-solution is placed in an open container. The vapor is in contact with the salt-solution.

[1] T_1: Changing an influence of evaporation:

I-[(amount-of ?liquid), (A (evaporation ?self))] →
I-[(amount-of (solvent-of ?liquid)), (A (evaporation-rate ?self))].

The evaporation process is modified so that it influences the amount of the solvent of the solution rather than the entire solution.

[2] T_2: Adding a new relation to evaporation:

(Q- (concentration ?liquid) (amount-of ?liquid)).

A negative qualitative proportionality between the concentration of the solution and the amount of the solution is added as a new relation to the evaporation process.

[3] T_3: Changing a relation of the solution:

(Q- (concentration ?solution) (amount-of (solvent-of ?solution))) →
(Q- (concentration ?solution) (amount-of ?solution)).

The individual view for the solution is modified by widening the scope of one of its relation. The concentration of the solution is made qualitatively proportional to the amount of the solution instead of the amount of the solvent of the solution.

The three revised theories can be compared based on the structural simplicity criterion. Theories T_1 and T_3 involve modifications to the existing components of the theory whereas theory T_2 involves the

addition of a new component to the theory. According to the metric described above, theories T_1 and T_3 are simpler than theory T_2 because each has one fewer components. Therefore, the structural simplicity criterion prefers theories T_1 and T_3 to theory T_2.

7.3.2. The Simplicity of the Explanations Constructed by the Theory

This criterion is estimated by 1) collecting the explanations constructed by each theory for the same set of observations, and 2) comparing the complexity of the explanations to identify the theory that provides the simplest explanations for the set of observations. In order for the estimate to be reliable a) a large number of observations are required, and b) at least two of the theories being compared must construct different explanations for each observation.

Note that each of the theories being compared is successfully tested using exemplar-based theory rejection. Therefore, each theory has associated with it an exemplar space consisting of exemplars of previous observations. One of the beneficial side-effects of constructing and maintaining the exemplar spaces is that they provide a rich source of observations that can be used to compute an estimate for the simplicity of the explanations constructed by the theories. The explanations for the exemplar observations were computed during the testing of the theory or were previously available from the exemplar space of the original theory. Therefore, the use of the exemplar observations to estimate the criterion does not involve any additional costly computation with respect to the construction of explanations for the observations.

The second condition for the reliable estimation of the criterion requires that at least two of the theories being compared construct different explanations for each observation. If the exemplar space is large then comparing the explanations constructed by each theory for each observation in the exemplar space will be prohibitively expensive. Instead, candidate observations are obtained from the exemplars that were retrieved and re-explained in order to test each theory using exemplar-based theory rejection. Figure 7.5 illustrates two revisions to a theory. The exemplars shown in the two regions of the exemplar space are retrieved by exemplar-based theory rejection to test the corresponding revisions. If the reconstructed explanation is different from the original explanation (which can be determined and noted during exemplar-based theory rejection) then the observation is selected as a candidate observation for the computation of the criterion.

Various syntactic measures can be used to estimate the complexity of an explanation. Some of these are the number of links in the explanation, the depth of the explanation, and the number of

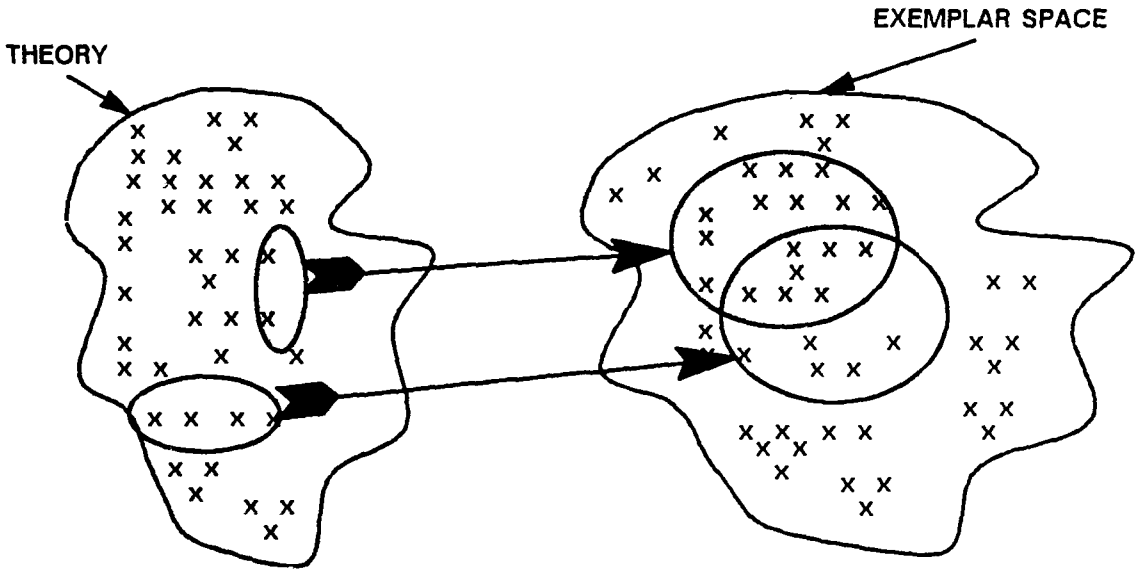


Figure 7.5 Two revised theories and the exemplars affected by each revised theory.

assumptions in the explanation. A metric for the complexity of an explanation that is based on the number of links in the explanation specifies a preference for theories that construct direct explanations for observations. A metric based on the depth of the explanation prefers theories that provide shallow explanations for observations. A metric based on the number of assumptions used in the explanation prefers theories that make less assumptions during the construction of explanations.

COAST measures the complexity of an explanation based on the number of links in the explanation. Figure 7.6 shows an explanation and the measure of its complexity. The metric specifies a preference for explanations that use direct influences rather than indirect influences (through qualitative proportionalities). This results in a preference for revised theories in which revisions are made to the influences rather than the relations of a process.

Explanation for (decrease (amount-of alcohol))

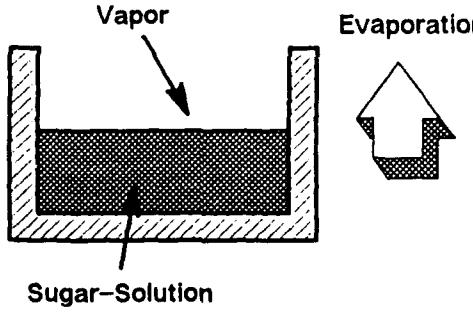
(decrease (amount-of alcohol))
I-[(amount-of alcohol), (A (evaporation-rate (evaporation alcohol vapor)))]
(active (evaporation alcohol vapor))
(open? (container alcohol))

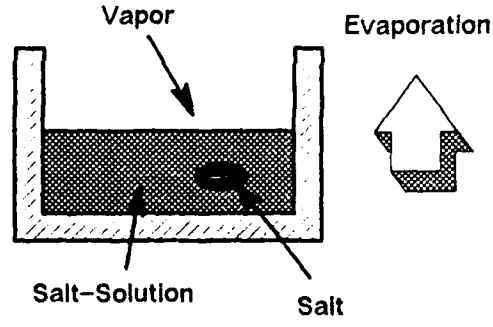
Complexity of explanation = Number of links = 4

Figure 7.6 The complexity of an explanation.

Example Illustrating the Comparison of Theories based on the Simplicity of the Explanations

The three revised theories, T_1, T_2 and T_3, introduced in the evaporation example of the previous subsection, are used to illustrate the comparison of theories based on the simplicity of the explanations constructed by each theory. To simplify the discussion, only two of the exemplar observations that are retrieved and re-explained during the testing of the revised theories by exemplar-based theory rejection are considered. The first exemplar observation is a decrease in the amount of the sugar-solution in the scenario shown in figure 7.7a. The explanation for this





Observation =
(decrease (amount-of sugar-solution))

(a)

Observation =
(increase (concentration salt-solution))

(b)

Figure 7.7 (a) Sugar-solution placed in an open container. Its amount is observed to decrease. (b) Salt-solution placed in an open container. Its concentration is observed to increase (and was originally explained as due to the dissolving of salt in the solution alone).

observation constructed by the theories T_2 and T_3 is simpler than that constructed by the theory T_1 (figure 7.8). Similarly, the explanation for the second exemplar observation of an increase in the concentration of the salt-solution in the scenario shown in figure 7.7b constructed by the theories T_2

and T_3 is simpler than that constructed by the theory T_1. Therefore, based on these two exemplars, the simplicity of explanations criterion prefers theories T_2 and T_3 to theory T_1. The explanations over a large number of exemplar observations are compared in this manner to determine which theory produces simpler explanations.

```
Explanations for (decrease (amount-of sugar-solution))

(decrease (amount-of sugar-solution))
   I-[(amount-of sugar-solution), (A (evaporation-rate (evaporation sugar-solution vapor)))]
      (active (evaporation sugar-solution vapor))
         (open? (container sugar-solution))

            Explanation complexity = 4
```

(a)

```
(decrease (amount-of sugar-solution))
   (decrease (amount-of (solvent-of sugar-solution)))
      I-[(amount-of (solvent-of sugar-solution)),
                        (A (evaporation-rate (evaporation sugar-solution vapor)))]
         (active (evaporation sugar-solution vapor))
            (open? (container sugar-solution))
   (Q+ (amount-of sugar-solution) (amount-of (solvent-of sugar-solution)))
      (active (solution sugar-solution))
         (greater-than (A (amount-of (solute-of sugar-solution))) 0)

            Explanation complexity = 8
```

(b)

Figure 7.8 (a) The explanation constructed based on theories T_2 and T_3. (b) The explanation constructed based on theory T_1.

7.3.3. The Predictive Power of the Theory

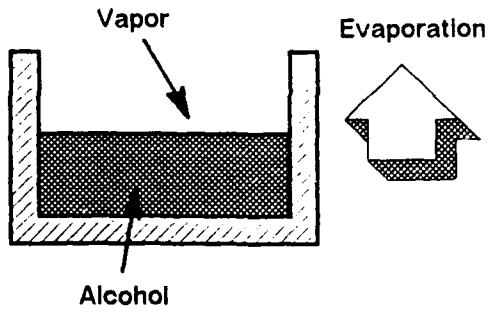
The estimation of the predictive power of a theory requires 1) a collection of scenarios 2) the computation of the behavior in each scenario predicted by each theory, and 3) a comparison of the predictive power of each theory for the scenarios. In order to reliably estimate the predictive power of each theory, a large collection of scenarios is required. The exemplar spaces associated with the theories provide a rich source of scenarios. The behavior of these scenarios is partially known (for example, the observation of the exemplar). However, each of the revised theories may make predictions that have not been previously observed. These predictions can be compared to identify theories with greater predictive power.

Some of the metrics that can be used for estimating the predictive power of a theory for a given scenario are the number of predictions made by the theory for the scenario, the number of predicted changes made by the theory for the scenario, and the number of predictions made by the theory for the scenario that can be verified through experimentation. A metric that uses the number of predictions made by a theory prefers revised theories that introduce new quantities and, therefore, new predictions about the values of the new quantities. A metric that uses the number of predicted changes made by a theory prefers revised theories that predict changes to quantities to revised theories that predict quantities remain constant. A metric that uses the number of predictions made by the theory which can be confirmed by experimentation prefers theories that can be easily falsified or corroborated.

COAST uses the number of predicted changes made by the theory for a scenario to estimate the predictive power of the theory for the scenario. Figure 7.9 shows the procedure for the computing the metric. Figure 7.10 shows a simple scenario, the changes for the scenario predicted by the theory that includes a process description for evaporation and the predictive power of the theory for the scenario.

```
Procedure Predictive-power-of-theory (theory)
   Select a set of exemplar scenarios
   ;;; The number of exemplar scenarios selected is pre-specified as a parameter.
   ;;; The scenarios are selected at random.
   predictive-power = 0
   For each selected scenario do
      Compute the behavior of the scenario
      predictive-power = number of quantities predicted to be increasing in the scenario
                       + number of quantities predicted to be decreasing in the scenario
```

Figure 7.9 The procedure for computing the predictive power of a theory.



Predicted Changes:
(increase (amount-of vapor))
(decrease (amount-of alcohol))

Predictive Power (for scenario) = 2

Figure 7.10 The predictive power of the theory for the given scenario.

Example Illustrating the Predictive Power of a Theory

The three revised theories, T_1, T_2 and T_3, of the evaporation example used in the previous subsections, are again used to illustrate the comparison of theories based on the predictive power of each theory. To simplify the discussion, only two exemplar scenarios are considered. Consider the exemplar scenario shown in figure 7.11a. The theory T_3 makes a number of predictions in addition to those made by the theories T_1 and T_2. It predicts that the concentration of the solution in the first container increases and the concentration of the solution in the second container decreases because the amounts of the two solutions decrease and increase respectively. However, since no evaporation process is active, the theories T_1 and T_2 predict that the concentrations remain the same. In the scenario shown in figure 7.11b, all three theories predict that the concentration of the solution in the first container increases. However, the theories T_1 and T_2 predict that the concentration of the second solution also increases due to the inflow of the higher concentration solution. According to the theory T_3, the concentration of the second solution is positively affected by the inflow of the first solution and negatively affected by the increase in the amount. Hence, depending on which dominates, the concentration can increase, decrease or remain constant. Based on these two scenarios, the theory T_3 is preferred since it has greater predictive power as compared to the theories T_1 and T_2. If there are more exemplar scenarios then

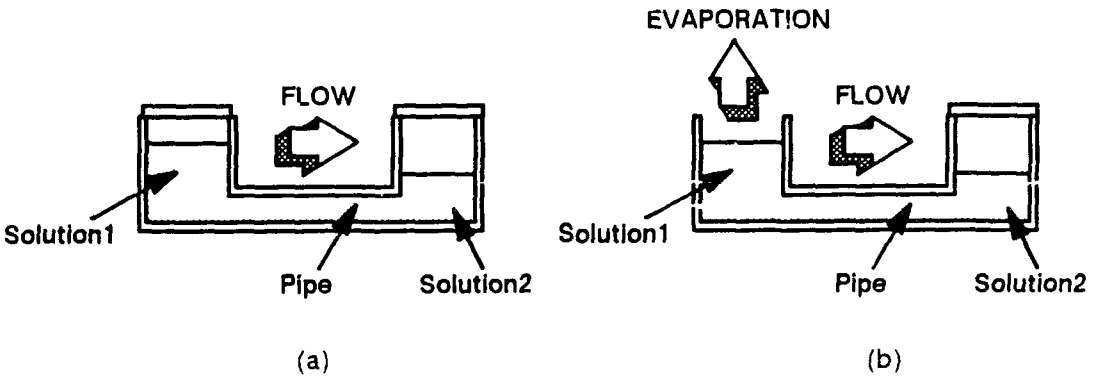


Figure 7.11 (a) A scenario in which two solutions of equal concentration are connected together by a pipe. The pipe permits flow and the pressure at the first solution is greater than the pressure at the second solution. The two solutions are placed in closed containers. (b) The container of the first solution is open and the concentration of the first solution is greater than the concentration of the second solution.

the predictive power can be computed as above and the theory with the greater predictive power can be selected.

7.3.4. Combining the Three Criteria

The estimates for each criterion are combined using a function that determines which of the three criteria (if any) play a dominant role. In scientific discovery, the simplicity of the explanations is of major consideration in the selection of a scientific theory from competing theories of equal explanatory power. Another important consideration is the predictive power of a theory. Its importance is due to the fact that a theory that makes a larger number of predictions can be more readily corroborated or refuted by experimental findings. Therefore, if the system is to provide a computational model for scientific discovery then the combination function must give greater importance to the estimates for the simplicity of the explanations constructed by the theory and the predictive power of the theory. If, however, the major concern of the system is the ease with which the theory can be manipulated or the facility with which the explanations provided by the theory can be assimilated (as is the case for practical systems) then the structural simplicity of the theory and the simplicity of the explanations computed by the theory must receive higher priority.

Figure 7.12 shows a simple procedure used by COAST to combine the three estimates The weight of each estimate can be set to reflect the importance of the estimate in the combination function.

```
Procedure combination-of-estimates
   Final estimate =
    weight-of-structural-simplicity * normalized-estimate-for-structural-simplicity +
    weight-of-simplicity-of-explanations * normalized-estimate-for-simplicity-of-explanations
    - weight-of-predictive-power * normalized-estimate-for-predictive-power
   ;;; The weights are user-supplied.
```

Figure 7.12 The procedure for computing the predictive power of a theory.

7.4. Discussion

Rose and Langley [Rose86] describe a different criterion for selecting the best hypothesis which is used by their system, STAHLP. STAHLP estimates the cost of making the modifications suggested by each hypothesis. It selects the one with the lowest cost on the basis that it will have the least impact on the existing beliefs of the system. The cost of making the revisions can also be used as a criterion in explanation-based theory revision. An estimate for the impact of the revisions on the beliefs of the system is the number of exemplar observations that had to be re-explained differently by each theory.

This chapter described three criteria to select a predetermined number of theories from a set of given theories. The three criteria are based on the structural simplicity of the theories, the simplicity of the explanations constructed by the theories and the predictive power of the theories. Metrics for measuring the complexity of a theory, the complexity of an explanation and the predictive power of a theory for a given scenario were defined. The selection of theories based on these criteria was illustrated with an example drawn from the liquids domain.

CHAPTER 8

ADDITIONAL APPLICATIONS OF EXPLANATION-BASED THEORY REVISION

8.1. Introduction

Previous chapters described how COAST extends its domain theory by learning new processes, learning new conditions and effects of existing processes, and correcting faulty conditions and effects of existing processes. This chapter focuses on two other applications of explanation-based theory revision. Experimentation-based hypothesis refutation can provide a partial solution to the multiple explanations problem – a central problem in employing explanation-based learning with imperfect domain theories. Explanation-based theory revision can provide a computational model for certain aspects of the scientific discovery process.

8.2. The Multiple Explanations Problem in Explanation-Based Learning

Explanation-based learning [DeJong86, Mitchell86] is a learning method in which the system is provided a theory of the domain and a training example. The system uses the theory to construct an *explanation* for why the training example is a member of the goal concept. This explanation is generalized into a general rule in such a manner that the constraints that were identified in the explanation are maintained. The general rule is applicable to conceptually similar examples. Explanation-based learning has been extensively researched in the past few years and a number of methodologies have been been developed for performing this type of learning [DeJong86, Hirsh87, Kedar-Cabelli87, Minton87, Mitchell86, Mooney86, O'Rorke87, Rosenbloom86]. It has been demonstrated in a wide variety of domains including natural language understanding [Mooney88], robotics [Segre87], game playing [Minton84], physics [Shavlik85], theorem proving [O'Rorke87], and circuit design [Ellman85, Mahadevan85].

One of the important problems in explanation–based learning is the *multiple explanations problem* [Rajamoney88b]: the explanation construction process yields many, incompatible explanations for why the given example is a member of the goal concept. A standard explanation–based learning system cannot cope with the multiple explanations problem. Such a system relies on a single explanation or multiple, but equally valid, explanations from the domain theory. For multiple valid explanations, the hope is that the arbitrary selection of an explanation will not have major implications on the learning process. Any selection results in a different, but correct, generalization. Multiple incompatible explanations pose a difficult problem. The system cannot arbitrarily select an explanation because the selection of an incorrect explanation can have profound implications on the subsequent problem solving and learning of the system. Nor can it generalize all the explanations because then the problem solving component will have difficulty in selecting the correct generalized rule.

As an example of the multiple explanations problem consider a robot that is operating in the real world. Since it is impossible to completely specify the state of the world, the robot must function with incomplete information. Suppose it observes that the length of a string attached to a wall is increasing. Due to the limited information the robot can construct three different explanations for the increase in the length of the string: 1) The string is hot and is expanding. 2) The string is growing with time like children do. 3) The string is elastic and is being pulled. If the robot were to arbitrarily select an explanation, then it may conclude incorrectly that all strings age and may never learn about elastic strings. Or if it generalizes all the explanations, then when it has to increase the length of a string it may elect to apply the first generalized rule. In which case, it may decide to heat the string.

8.2.1. Experimentation–based Hypothesis Refutation and the Multiple Explanations Problem

Experimentation–based hypothesis refutation provides a partial solution to the multiple explanations problem. Recall that experimentation–based hypothesis refutation is a method for eliminating competing hypotheses. It involves designing and conducting experiments to obtain additional data from the domain. If the experiments are suitably designed the obtained data will be inconsistent with the predictions of a number of hypotheses. These hypotheses can then be eliminated from further consideration.

The application of experimentation-based hypothesis refutation to the multiple explanations problem involves identifying the hypotheses underlying the explanations, designing experiments to test the hypotheses and eliminating hypotheses that make predictions that are incompatible with the experimental observations. Figure 8.1 shows the architecture of the experimentation-based

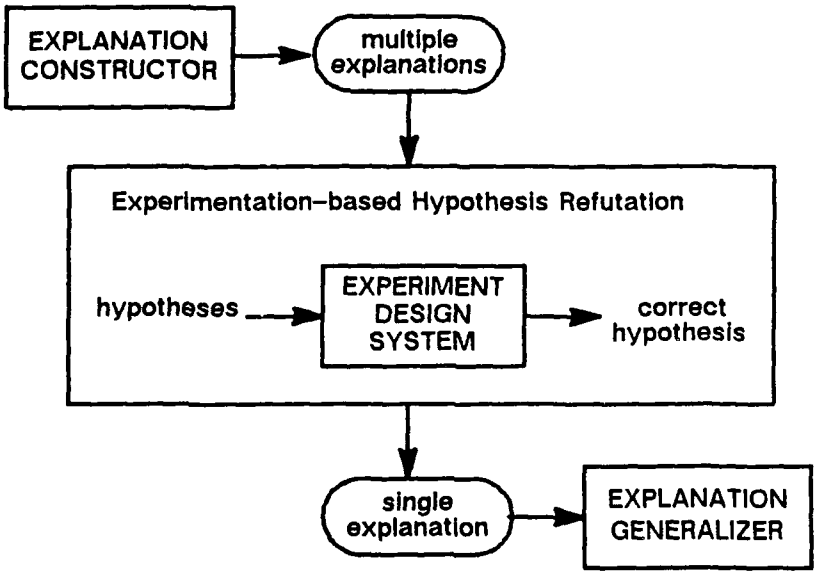


Figure 8.1 The architecture of the experimentation-based hypothesis refutation system for the multiple explanations problem.

hypothesis refutation system for the multiple explanations problem.

The hypotheses underlying each explanation are the hypotheses made by the system during the construction of each of the incompatible explanations. There are many situations in which the system must form hypotheses in order to construct explanations. Chapter 4 showed how a system operating with an incomplete and incorrect theory hypothesizes different revisions to the theory when failures occur. Multiple incompatible explanations can be constructed based on the different sets of revisions. When the system does not have sufficient information to uniquely determine which of many, mutually incompatible alternatives is correct, it may choose to hypothesize each alternative and construct an explanation based on each hypothesis. When the system operates with complex, intractable theories, it may hypothesize different approximations to the theory in order to construct the explanations. Only some of the approximations may be valid. Alternatively, the

system may refine an approximate explanation. It may obtain different refined explanations based on different hypotheses regarding the ranges or conditions of applicability of the approximation.

The identified hypotheses are provided to the experiment engine. The experiment engine designs experiments as described in section 5.2.1 to test the hypotheses. Experiments are designed based on an analysis of the predictions made by each hypothesis and according to the three strategies – elaboration, discrimination and transformation. Experimentation–based hypothesis refutation eliminates hypotheses that make predictions incompatible with the results of the experiments. The explanations based on these hypotheses are also discarded. If the experimentation method is complete and if the explanations given to the system can be distinguished from one other by experimentation then experimentation–based hypothesis refutation will identify the correct explanation. Otherwise it returns those explanations that are consistent with all the information obtained from the experiments. In this respect, experimentation–based hypothesis refutation is only a partial solution to the multiple explanations problem.

8.2.2. An Example Involving Multiple Explanations from an Intractable Theory

Doyle [Doyle86] describes an approach to the intractable theory problem in which the domain theory is represented at different levels of approximation. His system constructs explanations based on the tractable, approximate theory. If an explanation cannot be constructed the system reverts to the detailed theory to construct an explanation. The approximate theory is refined based on the constructed explanation. His system assumes that the detailed theory yields a single explanation. This assumption is not valid in most cases. In general, there will be a number of explanations for the failure and the correct explanation must be identified prior to the refinement of the approximate theory.

Consider an example from the domain of chemistry. COAST's domain theory includes process descriptions for the chemical decomposition of substances, the flow of heat, and the flow of electricity. The chemical decomposition of substances has two representations (figure 8.2): a simple process description of decomposition that ignores the influence of heat, electricity and catalysts, and a detailed representation that has three separate process descriptions for heat decomposition, catalytic decomposition and electrical decomposition.

```
Simple Decomposition:
   Individuals:
      ?substance ?product1 ?product2
   Preconditions:
      (decomposes ?substance)
   Quantity Conditions:
   Relations:
      rate Q+ amt(?substance)
      rate Q- amt(?product1)
      rate Q- amt(?product2)
   Influences:
      I+[amt(?product1), rate]
      I+[amt(?product2), rate]
      I-[amt(?substance), rate]
```

```
Heat Decomposition:
   Individuals:
      ?substance ?heat-flow
      ?product1 ?product2
   Preconditions:
      (decomposes-with-heat ?substance)
   Quantity Conditions:
      (active ?heat-flow)
      (> (A (temp ?substance)
         min-decomposition-temp))
   Relations:
      rate Q+ amt(?substance)
      rate Q- amt(?product1)
      rate Q- amt(?product2)
      rate Q+ rate(?heat-flow)
   Influences:
      I+[amt(?product1), rate]
      I+[amt(?product2), rate]
      I-[amt(?substance), rate]
      I-[heat(?substance), rate]
```

```
Catalytic Decomposition:
   Individuals:
      ?substance ?catalyst
      ?product1 ?product2
   Preconditions:
      (decomposes-with-catalyst
            ?substance ?catalyst)
      (in-contact ?substance ?catalyst)
   Quantity Conditions:
   Relations:
      rate Q+ amt(?substance)
      rate Q- amt(?product1)
      rate Q- amt(?product2)
      rate Q+ contact-area
            (?catalyst ?substance)
   Influences:
      I+[amt(?product1), rate]
      I+[amt(?product2), rate]
      I-[amt(?substance), rate]
```

```
Electrical Decomposition:
   Individuals:
      ?substance ?electrical-flow
      ?product1 ?product2
   Preconditions:
      (decomposes-with-electricity
            ?substance)
   Quantity Conditions:
      (active ?electrical-flow)
   Relations:
      rate Q+ amt(?substance)
      rate Q- amt(?product1)
      rate Q- amt(?product2)
      rate Q+ rate(?electrical-flow)
   Influences:
      I+[amt(?product1), rate]
      I+[amt(?product2), rate]
      I-[amt(?substance), rate]
```

Figure 8.2 The simple and the detailed theories of chemical decomposition.

COAST is shown a scenario in which oxygen is produced from water (figure 8.3). The system fails to construct an explanation for the generation of oxygen based on its simple theory of chemical decomposition because water does not ordinarily decompose. It then applies the more detailed theory and constructs the three different explanations shown in figure 8.4 corresponding to the decomposition of water due to heating, the decomposition of water due to the electric current and the decomposition of water due to the presence of platinum which serves as a catalyst.

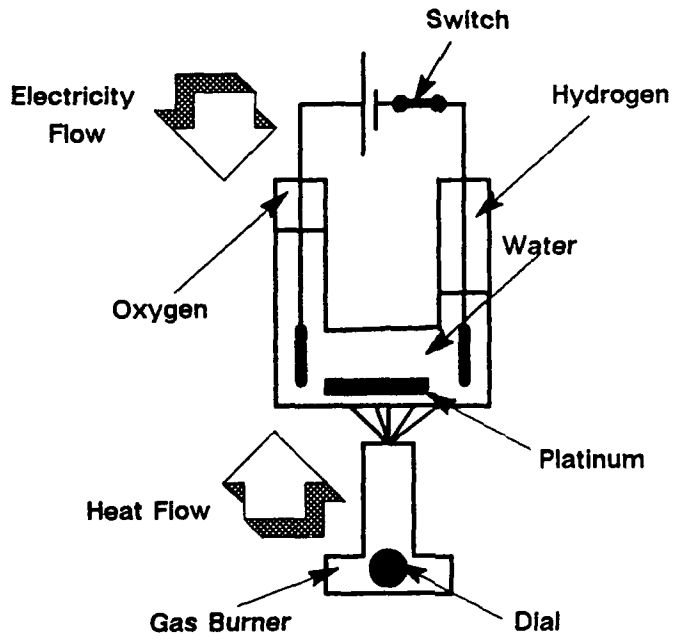


Figure 8.3 The scenario in which oxygen is generated.

```
(increase (amt oxygen))
   (I+ (amt oxygen) catalytic-decomposition-rate))
      (active (catalytic-decomposition water platinum oxygen hydrogen))
         (decomposes-with-catalyst water platinum)
            :Hypothesis
         (in-contact water platinum)

(increase (amt oxygen))
   (I+ (amt oxygen) heat-decomposition-rate)
      (active (heat-decomposition water <heat-flow1> oxygen hydrogen))
         (decomposes-with-heat water)
            :Hypothesis
         (active <heat-flow1>)
            <explanation1>
         (greater-than (A (temp water)) (A (min-decomposition-temp water)))

(increase amt oxygen)
   (I+ (amt oxygen) electrical-decomposition-rate)
      (active (electrical-decomposition water <electricity-flow1> oxygen hydrogen))
         (decomposes-with-electricity water)
            :Hypothesis
         (active <electricity-flow1>)
            <explanation2>
```

Figure 8.4 The three explanations for the observed increase in the amount of oxygen.

Note that simpler approaches to the multiple explanations problem will not work. Suppose the system refines the approximate theory of decomposition based on an arbitrarily selected explanation. If it selects the catalytic decomposition explanation instead of the correct electrical decomposition explanation, then the system will attempt to generate oxygen by adding a piece of platinum to water and will fail. Or suppose the system forms a new process based on a conjunction of the hypotheses. Then the system will be able to generate oxygen but will fail for other products formed by electrical decomposition if the original substance is destroyed by heating or mixing with platinum. Furthermore, the system must do additional work to achieve unnecessary goals such as generating a heat process and obtaining platinum. Combining all the hypotheses into a disjunctive precondition for a new process is not a satisfactory solution either. Each process has characteristics that are not shared by the others – for example, the rate of the decomposition is proportional to the heat supplied in the case of heat decomposition. This is not true of catalytic or electrical decomposition. Therefore, it is essential to identify the correct explanation.

COAST retrieves the three hypotheses underlying each of the explanations: water decomposes with heat, water decomposes with electricity and water decomposes in the presence of platinum. These hypotheses are tested by the experiment engine. Elaboration and discrimination in the original scenario fail because all the predictions obtained from the inference engine are supported by the three hypotheses. The experiment engine applies a transformation operator to construct a new scenario in which the heat from the burner is increased. The heat decomposition hypothesis predicts that oxygen is generated at a higher rate in the new scenario because the rate of the heat decomposition is proportional to the heat supplied. However, the other two hypotheses predict that the rate will remain the same since the decomposition rate is not affected by heat. A differential discrimination experiment that compares the rate at which oxygen is generated in each scenario is constructed. Suppose the rate is observed to be the same in both scenarios when the experiment is performed by an external agent. Based on this result, COAST eliminates the heat decomposition hypothesis and the corresponding explanation for the generation of oxygen. The experiment engine applies another transformation operator to construct a scenario in which the electrical switch is off – thereby, stopping the electrical current. The electrical decomposition hypothesis predicts that the generation of oxygen will stop in the new scenario. The catalytic decomposition hypothesis predicts that the generation of oxygen will continue undisturbed because catalytic decomposition is not affected by the flow of electric current. Suppose the production of oxygen is observed to stop

when the experiment is performed. Based on this experimental result, COAST eliminates the catalytic decomposition hypothesis and returns the electrical decomposition explanation as the correct explanation for the observed increase in the amount of oxygen.

8.2.3. Discussion of Related Work Addressing the Multiple Explanations Problem

The solution to the multiple explanations problem described above involves designing and conducting experiments to gather additional information and eliminating those explanations that are inconsistent with the obtained information. Dietterich and Flann [Dietterich88] describe an alternative approach called Induction Over Examples (IOE) that addresses the multiple explanations problem. IOE obtains all the explanations that can be constructed by the domain theory for each of the several training examples given to the system. Similarity-based learning techniques are applied to these explanations to obtain a single, general explanation for all the training examples. Since some of the explanations in the training examples are incorrect, the resulting generalized explanation can also incorporate some of the incorrect information. In contrast, experimentation-based hypothesis refutation designs experiments to test the ramifications of the hypotheses underlying each explanation. The additional information obtained through experimentation is used to eliminate the incorrect explanations and identify the correct explanation. Pazzani [Pazzani88b] describes a method for selecting the "best" explanation from a set of inconsistent explanations. In his method, the plausibility of each explanation is rated based on heuristics such as "favor hypotheses which account for a large number of observations" and "penalize hypotheses which violate a general theory of causality". Experimentation-based hypothesis refutation does not use heuristics to eliminate hypotheses. Only incorrect explanations, which are incompatible with the experimental results, are eliminated by experimentation. In contrast, heuristics, such as the ones described above, can potentially eliminate implausible, but correct, explanations. Heuristics can be advantageously used in conjunction with experimentation based hypothesis refutation to select the "best" explanation from the explanations remaining after experimentation.

8.3. Scientific Discovery

Research in scientific discovery addresses the problem of constructing computational models of the scientific discovery process. Scientific discovery is complex and multifaceted. The development of scientific theories occurs over a number of stages. The early stages involve the formulation of

empirical laws that summarize or describe the available data. Typical examples of this stage from the history of science include the formulation of Kepler's laws of planetary motion, the ideal gas law, and Dalton's law of multiple proportions. A number of systems such as BACON, DALTON, STAHL, GLAUBER [Langley86], ABACUS [Falkenhainer86], STAHLP [Rose86], and NGALUBER [Jones86] have been developed to model this stage in scientific discovery.

The later stages in scientific discovery are concerned with the development of rich theories that have considerable explanatory and predictive power. Theory development is cyclic in nature – theories are formed to explain existing data or empirical laws; the theories are constantly extended or modified to assimilate new anomalous observations; and lastly, the theories are radically changed or completely replaced by new theories (a paradigm shift in the terminology of Kuhn [Kuhn70]) when they become too unwieldy and unaesthetic.

While the early stages of scientific discovery focus on formulating qualitative and quantitative laws that describe and summarize the data, theory formation focuses on formulating new theories that explain the collected data and make predictions beyond the known data. Examples of theory formation from the history of science include the caloric theory which explained the flow of heat, the phlogiston and oxygen theories which explained combustion and related chemical reactions, and the "fluid" theories of electricity that explained electrical conduction. Falkenhainer [Falkenhainer87a, Falkenhainer87b] describes a computational model of scientific discovery called *verification-based analogical learning*. In his model, qualitative theories of the physical world are formed by analogy to known theories from other domains.

Theory revision deals with the extension or modification of existing theories to assimilate anomalous observations. The history of science provides numerous examples of theory revision: a number of changes were made to the Ptolemaic theory of astronomy to account for discrepancies between the observed motion and the predicted motion of the planets; the phlogiston theory in chemistry underwent a series of modifications to account for anomalous observations pertaining to chemical reactions and combustion; the caloric theory of heat was extended many times as additional data regarding the nature of heat was collected.

Theory revision differs considerably from theory formation. The existence of a theory that was successfully applied to explain previous observations has noteworthy implications for the process of theory revision. This theory has significant impact on every aspect of theory revision – the

identification of the anomalies, the determination of the appropriate extensions or modifications to the theory, the testing of competing revised theories and the selection of a best theory from revised theories of equivalent explanatory power.

8.3.1. Explanation–based Theory Revision: A Model for Scientific Theory Revision

Explanation–based theory revision provides a computational model for the theory revision process in scientific discovery. It models five stages of theory revision: detecting anomalous observations, proposing revised theories, designing experiments to test the proposed theories, rejecting theories that are not consistent with previously collected observations, and selecting a "best" theory from theories of equivalent explanatory power. The components of the model are:

[a] Identification of anomalous observations: Scientific theory revision commences with anomalous observations – observations that violate the expectations of the theory. Explanation–based theory revision detects anomalous observations when the theory fails to construct an explanation for the observations or when the predictions of the theory are not compatible with the observations.

[b] Proposing revised theories: The theory is augmented or modified to assimilate the anomalous observations. Explanation–based theory revision produces revised theories, that can explain the anomalous observations, according to the procedure described in chapter 4. The procedure constrains the space of possible revised theories by exploiting knowledge about the initial theory, the anomalous observations, the scenario in which the anomalous observations were encountered and the construction of explanations for the anomalous observations.

[c] Testing theories by experimentation: Experimentation is a central component in the scientific discovery process. Experiments are designed and conducted to test the predictions of a theory and to eliminate competing theories. Experimentation–based hypothesis refutation provides a computational model for the role of experiments in the elimination of competing theories. Experiments are designed by analyzing the predictions of the proposed theories and according to the three strategies – elaboration, discrimination and transformation – described in chapter 5.

[d] Testing theories by exemplar re-explanation: Revised theories that cannot account for previously collected data are untenable. Exemplar-based theory rejection tests whether the proposed theories are consistent with the collected data. Theories that cannot re-explain selected observations which were previously encountered are rejected.

[e] Selection of a "best" theory: When a number of theories account for the same collection of known data the selection of a theory is based on aesthetic criteria. Explanation-based theory revision uses three aesthetic criteria to select a theory: simplicity of the theory, simplicity of the explanations provided by the theory (based on the exemplar explanations) and the predictive power of the theory (based on the predictions for the exemplar scenarios). The second criterion corresponds to the Occam's razor criterion that is commonly used in scientific discovery. The third criterion, preferring a theory that provides more predictions, is another commonly used criterion in scientific discovery [Popper68].

8.3.2. An Example: Revision of the Phlogiston Theory of Combustion

The theory revision model described above is implemented in the COAST system and is illustrated with an example involving revisions to the *phlogiston* theory – a theory that was widely believed in chemistry during the 17th and 18th centuries. The phlogiston theory was proposed by the early chemists to explain combustion. According to the theory, when a substance burns it decomposes into a residue and an entity called *phlogiston*. Phlogiston manifests itself as the smoke, fire, heat and light associated with combustion. The phlogiston theory was revised a number of times and eventually could provide explanations for observations related to a large number of chemical reactions, combustion of substances and phenomena associated with heat such as the expansion and compression of gases and the formation of mixtures. This example describes how the phlogiston theory can be revised by explanation-based theory revision to account for anomalous observations concerning the change in weight of substances on combustion.

Figure 8.5 shows the naive representation of the phlogiston theory in the framework of QP theory used by COAST. Substances are composed of phlogiston and the residue remaining after combustion. The weight of a substance depends on its phlogiston and its residue. Combustion is represented as a process which negatively influences the phlogiston of a substance. The conditions for the combustion of a substance are: 1) The substance must be combustible, that is, it must be in

contact with a flame. 2) The phlogiston of the substance must be greater than its minimum phlogiston (the residual phlogiston in the substance after it has completely burned).

```
Phlogiston Theory:
    Substance(?substance)
      Individuals: ?substance
      Preconditions:
      Quantity Conditions: (greater-than (A (amount ?substance)) 0)
      Relations:  (greater-than (A (phlogiston ?substance)) 0)
                  (greater-than (A (residue ?substance)) 0)
                  (Q+ (weight ?substance) (phlogiston ?substance))
                  (Q+ (weight ?substance) (residue ?substance))
  Combustion (?substance)
      Individuals: ?substance
      Preconditions: (combustible ?substance)
      Quantity Conditions:
         (greater-than (A (phlogiston ?substance)) (A (phlogiston-minimum ?substance)))
      Relations:
      Influences: I-[(phlogiston ?substance), (A (combustion-rate ?self))]
```

Figure 8.5 COAST's representation of the phlogiston theory in the framework of QP theory.

The phlogiston theory shown in figure 8.5 provides explanations for the simple observations about combustion that were made by the early chemists. One such observation was that the weight of the substance undergoing combustion decreases. The explanation constructed by the theory for a decrease in the weight of a piece of wood undergoing combustion is shown in figure 8.6.

```
Explanation1:
   (decrease (weight wood))
      (Q+ (weight wood) (phlogiston wood))
         (active (substance wood))
            (greater-than (A (amount wood)) 0)
      (decrease (phlogiston wood))
         I-[(phlogiston wood), (A (combustion-rate (combustion wood)))]
            (active (combustion wood))
               (combustible wood)
               (greater-than (A (phlogiston wood)) (A (phlogiston-minimum wood)))
      (greater-than (A (phlogiston wood)) 0)
         (active (substance wood))
            (greater-than (A (amount wood)) 0)
```

Figure 8.6 The explanation for the decrease in weight of wood on combustion.

When the methods of measurement became more refined, the chemists discovered that a number of substances, notably metals, *gained* weight during combustion. The phlogiston theory described in figure 8.5 cannot explain this observation. The theory must be extended to assimilate this

anomalous observation. Explanation-based theory revision proposes revisions to the theory based on the procedure described in chapter 4. Four of the concrete revisions to the phlogiston theory that can explain the anomalous observation are[1]:

(1) Inverting a relation of the substance definition:

(Q+ (weight ?substance) (phlogiston ?substance))
→ (Q- (weight ?substance) (phlogiston ?substance)).

In the revised theory, the weight of a substance is inversely proportional to the phlogiston of the substance. The explanation for the increase in weight of the metal based on the revised theory is shown in figure 8.7a.

(2) Inverting an influence of the combustion process:

I-[(phlogiston ?substance), (A (combustion-rate ?self))]
→ I+[(phlogiston ?substance), (A (combustion-rate ?self))].

In the revised theory, the combustion process positively influences the phlogiston of the substance undergoing combustion. The explanation for the increase in the weight of the metal based on the revised theory is shown in figure 8.7b.

(3) Deleting a relation from the substance definition:

(greater-than (A (phlogiston ?substance)) 0) →.

The revised theory does not impose any constraint on the value of the phlogiston of a substance. Substances may have positive phlogiston, zero phlogiston or negative phlogiston. The explanation for the increase in the weight of the metal based on the revised theory is shown in figure 8.7c. The explanation is based on the assumption that the phlogiston of the metal is negative.

(4) Adding a new influence to the combustion process:

→ I+[(residue ?substance), (A (combustion-rate ?self))].

In the revised theory, the combustion process positively influences the residue of the substance undergoing combustion. The explanation for the increase in the weight of the

[1] There are other revisions that are handled in a similar fashion.

metal based on the revised theory is shown in figure 8.7d. The explanation is based on the assumption that the increase in weight due to the increase in the residue of the metal dominates over the decrease in the weight due to the loss of phlogiston from the metal.

COAST uses exemplar-based theory rejection to test each of the four revised theories. Exemplar-based theory rejection checks whether each theory can explain the previously observed decrease in the weight of the piece of wood on combustion. The first revised theory cannot explain the decrease in the weight of the wood because, according to the revised theory, the weight of the wood is inversely proportional to the phlogiston of the wood and the phlogiston of the wood is decreasing due to combustion. The second revised theory also cannot explain the decrease in the weight of the wood because, according to the revised theory, the phlogiston of the wood is increasing due to the combustion and the weight of the wood is qualitatively proportional to the phlogiston of the wood. The third revised theory can explain the decrease in the weight of the wood under the assumption that the phlogiston of the wood is positive. The explanation is shown in figure 8.8a. The fourth revised theory can also explain the decrease in the weight of the wood under the assumption that the increase in the weight of the wood due to the increase in the residue dominates the decrease in the weight of the wood due to the loss of the phlogiston. The explanation is shown in figure 8.8b. Exemplar-based theory rejection rejects the first two revised theories since they cannot explain one of the known observations.

The two revised theories – the theory obtained by deleting the condition that phlogiston must be positive and the theory obtained by augmenting the combustion process with a new influence – are of equal explanatory power since they account for all the known observations. COAST prefers the former theory to the latter since it has fewer components and provides simpler explanations (based on a count of the number of links in each explanation).

COAST successfully models the revision of the phlogiston theory since both these theories were entertained by the early chemists as extensions of the original phlogiston theory that could account for the observed increase in the weight of some substances due to combustion[2] [Kuhn70]. However, it is not clear if the early chemists preferred one theory to the other.

[2] The second theory is actually a closely related version of the theory proposed by the early chemists – that is, fireparticles enter the substance during combustion. In our representation, this corresponds to an increase in the residue of the substance.

```
Explanation 2 for Q+ → Q-
   (increase (weight metal))
      (Q- (weight metal) (phlogiston metal))
         (active (substance metal))
            (greater-than (A (amount metal)) 0)
      (decrease (phlogiston metal))
         I-[(phlogiston metal), (A (combustion-rate (combustion metal)))]
            (active (combustion metal))
               (combustible metal)
               (greater-than (A (phlogiston metal)) (A (phlogiston-minimum metal)))
      (greater-than (A (phlogiston metal)) 0)
            (active (substance metal))
               (greater-than (A (amount metal)) 0)
```

(a)

```
Explanation 3 for I- → I+
   (increase (weight metal))
      (Q+ (weight metal) (phlogiston metal))
         (active (substance metal))
            (greater-than (A (amount metal)) 0)
      (increase (phlogiston metal))
         I+[(phlogiston metal), (A (combustion-rate (combustion metal)))]
            (active (combustion metal))
               (combustible metal)
               (greater-than (A (phlogiston metal)) (A (phlogiston-minimum metal)))
      (greater-than (A (phlogiston metal)) 0)
            (active (substance metal))
               (greater-than (A (amount metal)) 0)
```

(b)

```
Explanation 4 for the deleted (greater-than (A (phlogiston ?substance)) 0)
   (increase (weight metal))
      (Q+ (weight metal) (phlogiston metal))
         (active (substance metal))
            (greater-than (A (amount metal)) 0)
      (increase (phlogiston metal))
         I-[(phlogiston metal), (A (combustion-rate (combustion metal)))]
            (active (combustion metal))
               (combustible metal)
               (greater-than (A (phlogiston metal)) (A (phlogiston-minimum metal)))
      (less-than (A (phlogiston metal)) 0)
```

(c)

Figure 8.7 The explanations for the increase in the weight of the metal by each revised theory.

```
Explanation 5 for the new influence I+[(residue ?substance), (A (combustion-rate ?self))]
   (increase (weight metal))
      (decrease (weight metal))
         (Q+ (weight metal) (phlogiston metal))
            (active (substance metal))
               (greater-than (A (amount metal)) 0)
         (decrease (phlogiston metal))
            I-[(phlogiston metal), (A (combustion-rate (combustion metal)))]
               (active (combustion metal))
                  (combustible metal)
                  (greater-than (A (phlogiston metal)) (A (phlogiston-minimum metal)))
            (greater-than (A (phlogiston metal)) 0)
               (active (substance metal))
                  (greater-than (A (amount metal)) 0)
      (increase (weight metal))
         (Q+ (weight metal) (residue metal))
            (active (substance metal))
               (greater-than (A (amount metal)) 0)
         (increase (residue metal))
            I+[(residue metal), (A (combustion-rate (combustion metal)))]
               (active (combustion metal))
                  (combustible metal)
                  (greater-than (A (phlogiston metal)) (A (phlogiston-minimum metal)))
            (greater-than (A (residue metal)) 0)
               (active (substance metal))
                  (greater-than (A (amount metal)) 0)
      (greater-than (increase (weight metal)) (decrease (weight metal)))
```

(d)

Figure 8.7 (continued) The explanations for the increase in the weight of the metal by each revised theory.

```
Explanation 6 for the deleted (greater-than (A (phlogiston ?substance)) 0)
   (decrease (weight wood))
      (Q+ (weight wood) (phlogiston wood))
         (active (substance wood))
            (greater-than (A (amount wood)) 0)
      (decrease (phlogiston wood))
         I-[(phlogiston wood), (A (combustion-rate (combustion wood)))]
            (active (combustion wood))
               (combustible wood)
               (greater-than (A (phlogiston wood)) (A (phlogiston-minimum wood)))
         (greater-than (A (phlogiston wood)) 0)
```

(a)

```
Explanation 7 for the new influence I+[(residue ?substance), (A (combustion-rate ?self))]
   (decrease (weight wood))
      (decrease (weight wood))
         (Q+ (weight wood) (phlogiston wood))
            (active (substance wood))
               (greater-than (A (amount wood)) 0)
         (decrease (phlogiston wood))
            I-[(phlogiston wood), (A (combustion-rate (combustion wood)))]
               (active (combustion wood))
                  (combustible wood)
                  (greater-than (A (phlogiston wood)) (A (phlogiston-minimum wood)))
            (greater-than (A (phlogiston wood)) 0)
               (active (substance wood))
                  (greater-than (A (amount wood)) 0)
      (increase (weight wood))
         (Q+ (weight wood) (residue wood))
            (active (substance wood))
               (greater-than (A (amount wood)) 0)
         (increase (residue wood))
            I+[(residue wood), (A (combustion-rate (combustion wood)))]
               (active (combustion wood))
                  (combustible wood)
                  (greater-than (A (phlogiston wood)) (A (phlogiston-minimum wood)))
            (greater-than (A (residue wood)) 0)
               (active (substance wood))
                  (greater-than (A (amount wood)) 0)
      (greater-than (decrease (weight wood)) (increase (weight wood)))
```

(b)

Figure 8.8 The reconstructed explanations for the combustion of wood exemplar.

8.3.3. Discussion of Related Work in Scientific Discovery

Explanation-based theory revision is proposed as a computational model for the process of theory revision in scientific discovery. However, the model has a number of limitations and is still a considerable distance from modeling all the subtleties and nuances of theory revision. History of

science has shown that a number of competing scientific theories are entertained prior to the dominance of a single theory [Kuhn70]. Each of these theories accounts for some of the known observations and either ignores the remaining anomalous observations or attempts to finesse them by *ad hoc* elaborations. Explanation-based theory revision cannot model this aspect of theory development since it zealously forces a theory to assimilate the anomalous observations. It does not permit the co-existence of competing theories of different explanatory power. Another major limitation stems from the qualitative nature of the representation and reasoning used by COAST. It cannot model revisions to quantitative theories nor can it design quantitative experiments to refute a competing theory. On a related note, the representation used by COAST must be extended to include other information like the manifestation of phlogiston in the smoke, fire, heat and light that accompanies combustion which appears to have played a major role in the acceptance of the phlogiston theory. Future work on the model will focus on addressing these limitations and will further test the model by attempting to replicate other episodes of theory revision from the history of science. In addition, work is in progress to form a unified framework of theory development that encompasses theory formation, theory revision and experimentation [Falkenhainer88b].

8.4. Summary

This chapter has described two additional applications of the methods developed in the previous chapters. Experimentation-based hypothesis refutation was proposed as a partial solution to the multiple explanations problem in explanation-based learning. Experimentation-based hypothesis refutation is invoked when an explanation-based learning system is confronted with multiple incompatible explanations. When invoked, the system identifies measurements and transformations of the world which would serve to disambiguate the alternative explanations.

Explanation-based theory revision was proposed as a computational model of theory revision in scientific discovery. Explanation-based theory revision provides methods for the detection of anomalous observations, the generation of revisions to a theory, the design of experiments to test competing theories, the rejection of theories that fail to explain previous observations and the selection of a "best" theory from the remaining theories of equivalent explanatory power.

CHAPTER 9

CONCLUSIONS

9.1. A Brief Review of Explanation–based Theory Revision

To recapitulate, explanation–based theory revision is a method for augmenting and correcting domain theories that are incomplete and incorrect. It consists of: 1) detecting problems with the theory by comparing the predictions based on the theory with the observations made from the domain; 2) generating theory revision hypotheses based on an analysis of the failure, the scenario in which the failure occurred and the explanations for the failure observation that can be constructed based on the hypotheses; 3) designing experiments to obtain additional information to eliminate incorrect hypotheses; 4) rejecting revised theories that cannot explain selected previous observations; and, 5) selecting a "best" theory from the remaining theories based on aesthetic criteria such as the structural simplicity of the theories, the simplicity of the explanations constructed by the theories, and the predictive power of the theories.

Explanation–based theory revision requires:

[a] An inference engine: The inference engine is employed to obtain the predictions made by a theory for a given scenario drawn from the domain. The theory input to the inference engine can be of three types: the failed theory, the failed theory augmented by constraints imposed by abstract theory revision hypotheses, and a revised theory constructed from the failed theory based on concrete theory revision hypotheses. The scenario input to the inference engine can also be of three types: the failure scenario, an experimentally constructed scenario, and an exemplar scenario.

[b] Observations from the domain: Explanation–based theory revision requires observations from the domain in order to identify problems with the theory. The observed behavior must be

input to the system in the form of qualitative changes for some quantities of a given scenario. In addition, experimentation-based hypothesis refutation requires an external agent to perform the designed experiments and input the experimental observations.

Apart from the above two requirements, explanation-based theory revision also requires various types of domain knowledge to guide the theory revision process:

[1] Theory: The domain theory plays an important role in each of the five steps: contradiction detection uses the theory to obtain predictions that can be compared with the observations made from the domain to identify problems with the theory; hypothesis generation uses the theory to constrain the types of revisions and the components to which the revisions are applied; experimentation-based hypothesis refutation uses the theory to obtain predictions to test the hypotheses; exemplar-based theory rejection uses the components of the theory to organize the exemplar space and identify the exemplars that are relevant for the testing of revised theories; and, the three criteria for selecting the best theory are computed using each of the competing revised theories.

[2] Scenario: The layout of the scenario is required to compute the predicted behavior. In addition, experimentation-based hypothesis refutation uses the layout of the scenario to obtain transformed scenarios.

[3] Incompatible values: Explanation-based theory revision requires the incompatible values of a quantity to be specified. This information is used by contradiction detection to identify problems with the theory, and by experimentation-based hypothesis refutation and exemplar-based theory rejection to refute hypotheses.

[4] Theory revision operators: Hypothesis generation requires a complete and correct set of theory revision operators. These operators perform the specified revisions to the failed theory to produce the revised theories.

[5] Abstraction and refinement methods: Hypothesis generation requires the knowledge to form and refine abstract hypotheses in order to structure the space of theory revision hypotheses.

[6] Explanation construction: Hypothesis generation requires knowledge about the general types of explanations in order to limit the hypotheses generated to those that can explain the observation that led to the failure.

[7] Experimental information about quantities: The quantities that are measurable, easily measurable and manipulable in a scenario must be specified.

[8] Scenario transformation operators: A set of scenario transformation operators must be specified. These operators are used by experimentation-based hypothesis refutation to construct new scenarios in which to perform experiments.

The next section discusses the limitations of explanation-based theory revision and COAST, and identifies promising areas for future research. The third section discusses previous research related to explanation-based theory revision. The fourth section outlines some practical applications of explanation-based theory revision. The fifth section comments on the major contributions of the thesis. Finally, the sixth section presents a brief summary of the chapter.

9.2. Limitations and Future Work

Some of the limitations of explanation-based theory revision and the COAST system are described below. Potential areas for further research are also identified.

[a] In explanation-based theory revision, an unfavorable sequence of anomalous observations can result in an unaesthetic theory – a theory that produces complicated explanations or is too unwieldy to use. One solution is to supplement explanation-based theory revision with a method for globally restructuring the theory. Such a method might keep track of all the anomalies and propose changes to the theory or form a new theory that can explain all the observations. Alternatively, the method may reformulate the theory obtained through explanation-based theory revision by collapsing components, deleting unnecessary components and re-arranging some of the components of the theory. This type of global re-structuring of the theory involves a large shift in perspective and has been termed as a *paradigm shift* [Kuhn70]. Some initial research in developing computational methods for restructuring theories is described in [Thagard88].

[b] Explanation-based theory revision has currently been attempted only on theories that provide descriptive explanations for the observed phenomena. For example, when an anomalous

decrease in the temperature of an evaporating liquid is observed, explanation-based theory revision augments the process definition of evaporation to include a new influence. The observed phenomenon is explained as a consequence of evaporation. However, a deeper explanation exists at the molecular level of description: the temperature of the evaporating liquid drops due to the loss in the total energy of the liquid, which, in turn, is due to the escape of high energy molecules from the surface of the liquid. A promising area of future research is the investigation of the potential application of explanation-based theory revision to inventing new levels of description. One approach might be to represent theories at a descriptive level (corresponding to process descriptions of evaporation in terms of the effect on macro-quantities such as temperature, amount of the liquid etc.) and a deeper explanatory level (corresponding to the molecular level description of evaporation in terms of the effect of the escape of high-energy molecules from the surface); and, to explicitly represent the relationship of the quantities in each level (such as the temperature of the liquid to the energy of the molecules of the liquid). Such an approach might be able to transform revisions and explanations in one theory to appropriate revisions and explanations in the other theory.

[c] Explanation-based theory revision has been demonstrated only for qualitative theories. Qualitative reasoning restricts the values of the quantities to a small, finite set. One extension to this type of reasoning is reasoning about the order of magnitude of quantities [Raiman86]. Incorporating order of magnitude reasoning into explanation-based theory revision can considerably enhance the power of explanation-based theory revision. For example, more types of failures can be detected and the hypotheses can be more effectively discriminated. However, the enhancement due to the extended set of values is at the expense of the added complexity to the detection of problems with the theory and the generation and the testing of hypotheses.

[d] Explanation-based theory revision has been demonstrated only for theories of physical domains. An area of future research is to apply the method for theories of non-physical domains such as narrative understanding. Quantities in such domains may correspond to the thematic goals of an agent. Experimentation in such domains might involve posing focused questions.

[e] A problem related to theory revision is diagnosis, that is, attributing the discrepancies in the observed and predicted behavior to malfunctioning of a set of components of the artifact. An interesting area of future research would be to develop a methodology to distinguish between problems due to the theory and problems due to the malfunctioning components. Such a methodology would determine whether a particular discrepancy is more likely to be due to a problem with the theory than a malfunctioning of the component. The methodology might include a set of preliminary tests to be performed on the theory and the artifact to determine the cause of the discrepancy.

[f] The present implementation of explanation-based theory revision, the COAST system, deals only with the behavior of individual qualitative states. Consequently, COAST cannot reason about problems that propagate over qualitative states. A future research goal is to extend COAST to handle sequences of qualitative states or an *envisionment*. Apart from broadening the range of problems, this extension would also help in constraining the hypotheses generated and in designing experiments.

[g] Apart from forming a new precondition predicate for a new process, COAST presently does not augment the existing vocabulary of the properties of objects. For example, COAST does not learn concepts such as the temperature of a liquid or the concentration of solutions. Learning such concepts and refining their definitions by experimentation is an area of future research. Alternative methods for discovering such concepts are described in [Langley86, Lenat83].

[h] For complex real-world domains, the domain theory must be represented in a hierarchy of theories with varying levels of detail [Doyle86, Sacerdoti74, Stefik81]. This enables the system to reason at the appropriate level of representation without getting bogged down in unnecessary details. Though explanation-based theory revision addresses the multiple explanations problem arising from hierarchical theories (please refer to chapter 8 for details) it does not address the additional problems of learning in the context of hierarchical theories. These problems are important topics for future research.

9.3. Related Work

ADEPT [Rajamoney85, Rajamoney86a, Rajamoney86b], COAST's predecessor, addressed the same problems as COAST. However, there are important differences in the two systems. ADEPT

proposes hypotheses for theory revision based on an analysis of the failed explanation structure. COAST, on the other hand, proposes hypotheses in a more general manner based on constraints imposed by various sources of knowledge. In addition, unlike ADEPT, it utilizes abstraction and refinement methods to structure the hypothesis space. ADEPT constructs experiments only from a pre-specified set of experiment classes. On the other hand, COAST relies on three general strategies for designing experiments: elaboration, discrimination and transformation. Consequently, COAST's experiment engine is amenable to the testing of a wide variety of hypotheses. ADEPT, unlike COAST, did not perform exemplar-based theory rejection or select a best theory based on aesthetic criteria. Furthermore, the representation for the domain theory adopted by ADEPT was *ad hoc* in many respects. Consequently, ADEPT's inference engine was considerably limited in its capabilities. COAST, on the other hand, adopts a well-specified, general representation language, Qualitative Process theory, for representing domain theories. COAST has been demonstrated on many examples from different domains including the liquids domain and chemistry.

Other research related to explanation-based theory revision falls into four broad categories: imperfect theory problems in explanation-based learning, theory revision in scientific discovery, learning of qualitative theories, and diagnosis.

Explanation-Based Learning

There has been considerable research in integrating empirical learning methods with explanation-based learning methods to revise incomplete and incorrect domain theories. Pazzani [Pazzani88a] describes a system called OCCAM that uses similarity-based techniques to acquire and revise the domain theory that is required for explanation-based learning. Lebowitz and Danyluk [Danyluk87, Lebowitz86a, Lebowitz86b] also integrate similarity-based and explanation-based learning methods to deal with incomplete theories. Lebowitz [Lebowitz86a, Lebowitz86b] uses causal knowledge that can be learned by similarity-based techniques to guide the construction of explanations in incomplete domains. Kodratoff and Tecuci [Kodratoff87a, Kodratoff87b] describe a learning apprentice system called DISCIPLE that combines explanation-based, analogy-based and similarity-based methods. DISCIPLE relies on similarity-based methods and intelligent interaction with the user to add new rules to the domain theory and eliminate incorrect rules from the domain theory.

In empirical learning methods [Dietterich83, Michalski83b, Mitchell78, Stepp86], training examples of a target concept are used to incrementally form descriptions of the concept. Explanation-based theory revision also includes an empirical component: exemplars form a set of training instances that guide the incremental revision of the theory. However, there are some important differences. In empirical learning methods, the training instances are positive or negative examples of the concept and are typically represented by a set of feature descriptors and values. On the other hand, in explanation-based theory revision, the exemplars are examples that illustrate the use of the components of the theory and are defined by the role played by the components in the construction of explanations for observations encountered by the system. In empirical learning methods, the similarities and differences in the features of the training instances are used to grow or refine concept descriptions. But, in explanation-based theory revision, exemplars are used to test revised theories. Revised theories that cannot construct explanations for the observations of exemplars affected by the revisions are rejected.

Laird [Laird88] describes a method for recovering from incorrect knowledge in SOAR [Laird87]. In SOAR, all knowledge is encoded in the form of productions. Therefore, incorrect productions are the only source of incorrect knowledge in SOAR. Laird's method for recovering from incorrect knowledge involves modifying the decisions in which the incorrect knowledge is used instead of modifying the incorrect productions themselves. The modified decisions prevent the selection of the incorrect knowledge. The method results in the addition of productions that correct the decisions. However, Laird's method does not address the problem of oscillation: since the corrected productions can themselves be modified by the same procedure, it is possible that the method may result in an endless sequence of modifications as described in section 6.1. On the other hand, in explanation-based theory, oscillation is prevented by using exemplars to reject proposed theories that were previously revised.

Scientific Discovery

Scientific discovery addresses the problem of developing computational models for the discovery process. Traditional research in scientific discovery has focused on the development of systems such as BACON, GLAUBER, DALTON, and STAHL [Langley86] that discover empirical quantitative and qualitative laws which summarize the given data. However, unlike explanation-based theory revision, these systems do not form or revise rich theories that can explain the given data. More recently, Nordhausen and Langley [Nordhausen87] have developed a system called IDS which

discovers richer descriptions called *qualitative schemas* and uses these to guide the discovery of quantitative laws. IDS, like BACON, GLAUBER, DALTON and STAHL, is also data-driven. On the other hand, COAST is theory-driven. COAST relies on considerable domain knowledge to guide the process of theory revision.

More recent research has addressed the problems of theory formation and theory revision. Falkenhainer [Falkenhainer87a, Falkenhainer87b, Falkenhainer89] describes a method called *verification-based analogical learning* for theory formation and theory revision. Verification-based analogical learning relies on analogical inference from a knowledge base of analogous precedents to form new theories and revise existing theories. It uses simulation to test the validity of tne proposed theories. Unlike verification-based analogical learning, explanation-based theory revision generates revised theories using constraints obtained from the failure, the scenario, the explanation construction process and the structure of the domain theory. It designs experiments and uses exemplars to test the validity of the revised theories. The two methods are complementary and initial efforts to integrate them to provide a more comprehensive account of theory development are described in [Falkenhainer88b].

Learning Qualitative Models of Physical Domains

Forbus and Gentner [Forbus83] propose a theoretical framework that provides a computational account of human learning in physical domains. They propose four learning stages in this framework – 1) protohistories – a context-specific, summarization of observed phenomena, 2) causal corpus – a weak theory consisting of simple causal relations, 3) naive physics – a theory involving mechanisms of change and 4) expert models – a theory that includes general, domain-independent laws and is capable of resolving ambiguities inherent in qualitative models. Forbus and Gentner propose Gentner's Structure Mapping theory [Gentner83], an analogy-based learning method, as the primary mechanism for learning naive physics from the causal corpus. Explanation-based theory revision focuses on revising qualitative theories of the physical world and corresponds to the third stage in the framework proposed by Forbus and Gentner. However, explanation-based theory revision uses explanation-based and experimentation-based methods as the primary learning strategies.

Diagnosis

Diagnosis [Buchanan84, Davis84, de Kleer87, Genesereth84, Reiter87] addresses the problem of tracing the malfunctioning of an artifact to a set of faulty components of the artifact. Diagnosis has many commonalities with theory revision. Both theory revision and diagnosis commence with discrepancies between the observed behavior of an artifact and the predicted behavior. Theory revision assumes that the artifact is functioning correctly and revises the theory to conform with the observed behavior. Diagnosis assumes that the theory is correct and proceeds to locate the faults in the artifact. Both tasks require the generation and testing of hypotheses. However, there are considerable differences in diagnosis and theory revision that prevent the development and application of uniform methods to effectively deal with both tasks.

[a] Complexity Issues: There are two major differences in the two tasks: 1) Theory revision deals with the problems of incompleteness and incorrectness. Components of the theory can be missing or can be incorrect. However, diagnosis deals only with the incorrectness problem. Components of the artifact can be incorrect but it is assumed that all the components of the artifact are specified. For example, diagnosis does not allow for the possibility of an unspecified transistor as a potential cause of the malfunctioning of a circuit. This restriction results in a very substantial reduction in the complexity of hypothesis generation in diagnosis as compared with theory revision. Consider, for example, a theory revision problem which can be addressed by the addition of a new quantity condition to a process. If there are ten quantities changing in the scenario then the number of new quantity conditions that can be incorporated into the process is of the order of 10^6 (obtained by considering an inequality relation between any two quantities). 2) The primary task in diagnosis is to locate the fault or faults in the artifact. However, in theory revision, apart from identifying which components of the theory are incorrect, the components of the theory must also be revised appropriately to conform with the observations. Typically there is a large number of feasible revisions for each type of component. This introduces an additional dimension of complexity which is not encountered in diagnosis. For example, if there are ten components of a theory that can be revised and each component can be revised in ten different ways then the number of hypothesized revisions is of the order of 10^{10}. The increased complexity due to incompleteness and revision makes the problem of generating hypotheses considerably harder for theory revision. Multiple faults are unthinkable in theory revision except under very

restrictive assumptions. The difference in complexity severely limits the applicability of the methods developed for diagnosis to theory revision. For example, hypothesis generation in de Kleer and Williams' General Diagnostic Engine (GDE) [de Kleer87] involves generating a candidate space consisting of all possible sets of faulty components. In general, such a total generation is impossible in the case of theory revision. Even if multiple faults are not considered, the set of all possible revisions is still too large for any practical system to cope with. Strong constraints and assumptions such as those described in chapter 4 have to be incorporated into the generation of hypotheses for theory revision.

[b] General rules vs specific components: Diagnosis deals with hypotheses involving the malfunctioning of specific components of the artifact. These hypotheses are not applicable to other artifacts. But in theory revision, the hypotheses involve general rules with variables. Consequently, the hypotheses are applicable to scenarios other than the one in which the anomalous observations were encountered. This has considerable impact on the testing of the hypotheses. The test generation methods in diagnosis consist of simple probing at selected sites of the artifact to measure the values. On the other hand, the testing strategies for theory revision can be much more sophisticated since the hypothesized revisions are applicable to other scenarios as well. Experimentation can construct new scenarios (without re-using any of the components of the old scenario) to test the revised theories. Exemplars of previous observations can be retained and used to test the revised theories.

[c] Fault-based Modes: Fault-based diagnosis [Buchanan84] relies on a prestored set of fault modes for a device. Failures of the device are processed by determining and applying the fault mode for each failure. Fault-based methods cannot be used for theory revision since: 1) Unlike diagnosis, where it is possible to determine which parts of the device are likely to fail, in theory revision, it is not feasible to determine which parts of the theory are likely to fail *a priori*. 2) Unlike diagnosis, where a component of the device may fail repeatedly, in theory revision, once a particular fault with the theory has been repaired it will not recur.

9.4. Potential Applications

The research described in this thesis has a number of potential applications:

Building Robust Expert Systems

A major impediment in the building of expert systems is the know edge acquisition bottleneck: the task of acquiring sufficient knowledge of the domain to ensure good performance. The direct acquisition of knowledge from a domain expert is an arduous task. The designer of the expert system has the tedious chore of anticipating the rich variety of tasks and the wide range of situations for which the system must perform correctly. This is often impossible in complex real-world domains. The designer is forced to make approximations and assumptions. This results in the construction of fragile expert systems that can fail frequently. Explanation-based theory revision offers a promising approach to building robust expert systems. An operational theory of the domain can be constructed by "quick and dirty" methods. This domain theory can be gradually improved by explanation-based theory revision through experimental interaction with the domain when failures are encountered. This approach has the advantage of insulating the designer from the revisions that must be made when failures due to new unanticipated examples or incorrect knowledge in the domain theory are encountered.

Intelligent Computer-aided Instruction

Intelligent computer-aided instruction involves the application of artificial intelligence techniques to improve computer-aided instruction [Sleeman82, Wenger87]. If a computer is to instruct a student effectively then it must have a good model of the student's knowledge of the domain. This is necessary to form good problem sets; debug the students' solutions; identify areas in which the student is deficient; etc. Unfortunately, a model of the student's initial knowledge of the domain cannot be constructed and supplied to the computer since the model is not static. As the student progressively learns about the domain, his or her model of the domain changes. If the computer does not change its model correspondingly it will not be able tutor effectively. Explanation-based theory revision offers a potential solution to this problem. It may be employed to incrementally revise the model of the student's knowledge used by the computer based on the input responses of the student. Experimentation in this domain might involve the construction of problem sets such that the responses of the student to the problems can identify which of a set of revised models is correct.

Learning Apprentice Systems

Learning apprentice systems are of increasing importance in machine learning. Learning apprentice systems have been defined [Mitchell85] as "the class of *interactive* knowledge-based consultants that directly assimilate new knowledge by observing and analyzing the problem solving steps contributed by the users through the *normal* use of the system". Learning apprentice systems operating in real world domains face the problem of revising their domain theories when they are inadequate to solve the input task. In such a case, it is desirable to limit the interaction with the expert as far as possible. Consequently, it is not desirable to ask the expert to provide a solution to the task or to fix the theory. Explanation-based theory revision provides a potential solution for focusing the interaction with the expert. Explanation-based theory revision may be able to construct revised theories that can solve the input task. In this case, experimentation-based hypothesis refutation can determine the set of questions to be asked. The questions are such that the responses will discriminate among the competing theories. This results in focused interaction with the user to identify the correctly revised theory.

9.5. Significance of the Thesis

This thesis is significant for a number of reasons:

Hypothesis Generation:

This thesis demonstrates how knowledge can be advantageously used to constrain the size of the hypothesis space. The method of hypothesis generation described in this thesis utilizes knowledge about the failure, the scenario in which the failure occurred and the explanations for the failure to prune the hypothesis space. Another major contribution is in structuring the hypothesis space for effective testing. A description is provided of how abstract hypothesis spaces are formed and how the testing of the abstract hypothesis space prior to the testing of the refined, concrete hypothesis spaces considerably limits the number of hypotheses that have to be tested.

Experimentation:

One of the unique contributions of this thesis is a general methodology for learning by experimentation. A model of experiment design and an implementation are described. The model included three general strategies for designing experiments - elaboration, discrimination and transformation. An analysis of the fundamental issues involved in the design

of experiments – efficacy, efficiency, tolerance of unavailable data and feasibility – is also presented.

Exemplars:

A major novel contribution of this thesis is an investigation of some of the detrimental consequences of adding new knowledge and modifying existing knowledge on the explanations constructed using the previous knowledge. The thesis describes a method called *exemplar-based theory rejection* for eliminating proposed theories that are not consistent with the previous observations of the system. The method describes how examples called *exemplars* that illustrate the use of the components of the theory in explaining observations are collected and how exemplars relevant to the testing of the revised theory are retrieved. The method rejects revised theories that cannot re-explain the observations of the relevant exemplars. An analysis of the efficacy and efficiency of the method is also presented. Further, it is shown that, under certain conditions, a theory revision system employing exemplar-based theory rejection to test theories does not oscillate, that is, exemplar-based theory rejection prevents the theory revision from generating revised theories that previously required theory revision.

Selection of Theories:

A major contribution of the thesis is a systematic approach for evaluating a theory based on syntactic criteria. The thesis described three syntactic criteria for rating theories: the simplicity of the theory, the simplicity of the explanations constructed by the theory and the predictive power of the theory. This thesis showed how the exemplar spaces associated with theories provide a rich source of explanations and predictions for comparing theories.

Learning Rich Theories:

Machine learning researchers involved in knowledge-intensive learning have largely ignored the advances in the development of rich, well-defined representations in other areas of AI such as qualitative reasoning [de Kleer84, Forbus84a, Kuipers84, Williams84] and have persisted in adopting impoverished representations such as STRIPS-like operators. Such formalisms cannot easily represent the gradual changes or the complex interactions in the physical world. Consequently, such machine learning research has not addressed many of the complexities and interesting issues such as the complexity of hypothesis generation and

the design of interesting experiments. This thesis represents a major departure from the traditional machine learning research in that it adopts a rich, well-defined formalism as a vehicle for conducting learning research.

Performance Element:

Research in machine learning has often de-emphasized the role of the performance element in the learning process and in the evaluation of the learned knowledge. In many cases, the performance element is ill-defined and *ad hoc*. Too often, the knowledge that has been acquired through learning is not put to test in a challenging task. Consequently, the inadequacies of the learned knowledge and the limitations of the learning method are not fully revealed. But, in explanation-based theory revision, the performance element plays a major role: it identifies inadequacies in the existing knowledge (which includes the learned knowledge) by using it in the demanding task of explaining observations or predicting the behavior of scenarios drawn from the domain.

9.6. Summary

A brief review of explanation-based theory revision was presented. The domain knowledge required by explanation-based theory revision was also described. The limitations of explanation-based theory revision and COAST were described and potential areas for future research were identified. Explanation-based theory revision was compared with related research in explanation-based learning, scientific discovery, the learning of qualitative models and diagnosis. Three potential applications of explanation-based theory revision were described. Finally, the major contributions of the thesis were presented.

APPENDIX A

DETAILS OF THE OSMOSIS EXAMPLE

A.1. Introduction

Section 2.3 presented brief descriptions of examples of how COAST revises an incomplete and incorrect theory of liquids. In that section, COAST is shown to acquire a new process to eliminate a failure. Three subsequent failures led to the refinement of the initially acquired description of the new process. This appendix presents a more detailed, annotated trace of COAST's behavior on each of the four learning episodes.

A.2. The Initial Theory

The initial theory given to COAST consists of six processes: evaporation of liquids, condensation of vapor, absorption of liquids by solids, release of absorbed liquids by solids, transfer of solid substances, and flow of fluids. The theory also includes an individual view describing solutions. The descriptions for each of these is shown below:

```
(tr-defprocess (solution ?solution)
    individuals          ((?solution (contained-liquid ?solution)))
    preconditions        ((soluble? (solute-of ?solution) (solvent-of ?solution)))
    quantityconditions   ((:greater-than (a (amount-of (solute-of ?solution))) 0))
    relations            ((Q+ (concentration ?solution) (amount-of (solute-of ?solution)))
                          (Q- (concentration ?solution) (amount-of (solvent-of ?solution)))
                          (Q+ (amount-of ?solution) (amount-of (solvent-of ?solution))))
    influences           ())

(tr-defprocess (evaporation ?liquid ?vapor)
    individuals          ((?liquid (contained-liquid ?liquid))
                          (?vapor (contained-gas ?vapor) (connection ?liquid ?vapor)))
    rate                 (evaporation-rate ?self)
    preconditions        ((open-container (container ?liquid)))
    quantityconditions   ()
    relations            ((Q+ (evaporation-rate ?self) (contact-area ?liquid ?vapor)))
    influences           ((I- (amount-of ?liquid) (a (evaporation-rate ?self)))
                          (I+ (amount-of ?vapor) (a (evaporation-rate ?self)))))
```

```
(tr-defprocess (condensation ?vapor ?liquid)
    individuals             ((?vapor (contained-gas ?vapor))
                             (?liquid (contained-liquid ?liquid) (connection ?liquid ?vapor)))
    rate                    (condensation-rate ?self)
    preconditions           ((open-container (container ?liquid)))
    quantityconditions      ()
    relations               ((Q+ (condensation-rate ?self) (contact-area ?liquid ?vapor)))
    influences              ((I+ (amount-of ?liquid) (a (condensation-rate ?self)))
                             (I- (amount-of ?vapor) (a (condensation-rate ?self)))))

(tr-defprocess (absorption ?solid ?liquid)
    individuals             ((?solid (solid ?solid) (connection ?solid ?liquid))
                             (?liquid (contained-liquid ?liquid)))
    rate                    (absorption-rate ?self)
    preconditions           ((absorbent ?solid))
    quantityconditions      ()
    relations               ((Q+ (absorption-rate ?self) (contact-area ?solid ?liquid)))
    influences              ((I- (amount-of ?liquid) (a (absorption-rate ?self)))))

(tr-defprocess (release ?solid ?liquid)
    individuals             ((?solid (solid ?solid) (connection ?solid ?liquid))
                             (?liquid (contained-liquid ?liquid)))
    rate                    (release-rate ?self)
    preconditions           ((absorbent ?solid))
    quantityconditions      ()
    relations               ((Q+ (release-rate ?self) (contact-area ?solid ?liquid)))
    influences              ((I+ (amount-of ?liquid) (a (release-rate ?self)))))

(tr-defprocess (add-solute ?solute-source ?solute-destination)
    individuals             ((?solute-source (contained-solid ?solute-source))
                             (?solute-destination (contained-solid ?solute-destination)
                                    (transfer-connection ?solute-source ?solute-destination)))
    rate                    (add-solute-rate ?self)
    preconditions           ((transferable? ?solute-source ?solute-destination))
    quantityconditions      ()
    relations               ()
    influences              ((I- (amount-of ?solute-source) (a (add-solute-rate ?self)))
                             (I+ (amount-of ?solute-destination) (a (add-solute-rate ?self)))))

(tr-defprocess (fluid-flow ?source ?destination ?path)
    individuals             ((?source (contained-fluid ?source))
                             (?destination (contained-fluid ?destination))
                             (?path (path ?path) (path-connection ?source ?destination ?path)))
    rate                    (fluid-flow-rate ?self)
    preconditions           ((fluid-flow-aligned ?path))
    quantityconditions      ((:greater-than (a (pressure ?source))(a (pressure ?destination))))
    relations               ((Q+ (fluid-flow-rate ?self) (pressure ?source))
                             (Q- (fluid-flow-rate ?self) (pressure ?destination))
                             (Q- (fluid-flow-rate ?self) (LENGTH ?path))
                             (Q+ (fluid-flow-rate ?self) (cross-sectional-area ?path)))
    influences              ((I+ (amount-of ?destination) (a (fluid-flow-rate ?self)))
                             (I- (amount-of ?source) (a (fluid-flow-rate ?self)))))
```

A.3. Explaining Observations Successfully

COAST uses the given theory to explain observations made in scenarios drawn from the liquids domain. These successfully explained observations are stored by COAST as exemplars for the different components of the theory.

A.3.1. A Scenario Involving a Flow of a Fluid

COAST is given a scenario in which a liquid in one container is connected to a liquid in another container by a path. In this scenario, a flow of the liquid occurs because the path is aligned for the flow of fluids and a pressure difference exists between the two liquids. The amount of liquid at the source of the flow is observed to decrease and this observation is explained by COAST using the given theory.

```
(tr-defScenario fluid-flow-works-scenario
      individuals          (std-fluid1 std-fluid2 std-path1)
      facts                ((contained-fluid std-fluid1) (contained-fluid std-fluid2) (path std-path1)
                           (path-connection std-fluid1 std-fluid2 std-path1) (fluid-flow-aligned std-path1)
                           (:greater-than (a (pressure std-fluid1)) (a (pressure std-fluid2))))
      quantities           ((amount-of std-fluid1) (amount-of std-fluid2) (LENGTH std-path1)
                           (cross-sectional-area std-path1) (pressure std-fluid1) (pressure std-fluid2))
      transformable-parameters nil
      transformations nil))
>(explain-observation '(:decrease (ds (amount-of std-fluid1))) *fluid-flow-works-scenario*)

Explaining (DECREASE (DS (AMOUNT-OF STD-FLUID1))) in <S-1 FLUID-FLOW-WORKS-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW>)

  Explanation 1 for (DECREASE (DS (AMOUNT-OF STD-FLUID1)))
    (I- (AMOUNT-OF STD-FLUID1) (A (FLUID-FLOW-RATE (FLUID-FLOW STD-FLUID1 STD-FLUID2
    STD-PATH1))))
       (ACTIVE (FLUID-FLOW STD-FLUID1 STD-FLUID2 STD-PATH1))
          (GREATER-THAN (A (PRESSURE STD-FLUID1)) (A (PRESSURE STD-FLUID2)))
          (FLUID-FLOW-ALIGNED STD-PATH1)
NIL
```

A.3.2. A Scenario in which Flow Fails

COAST is given a second scenario which is similar to the one shown above. However, in this scenario, unlike the first scenario, the path connecting the two liquids is not aligned for the flow of fluids. COAST explains why the amount of the liquid remains constant.

```
>(explain-observation '(:constant (ds (amount-of std-fluid3))) user:*fluid-flow-fails1-scenario*)

Explaining (CONSTANT (DS (AMOUNT-OF STD-FLUID3))) in <S-1 FLUID-FLOW-FAILS1-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION>
<CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW>)
```

```
Explanation 1 for (CONSTANT (DS (AMOUNT-OF STD-FLUID3)))
  (I- (AMOUNT-OF STD-FLUID3) (A (FLUID-FLOW-RATE (FLUID-FLOW STD-FLUID3 STD-FLUID4
  STD-PATH2))))
     (INACTIVE (FLUID-FLOW STD-FLUID3 STD-FLUID4 STD-PATH2))
        (NOT (FLUID-FLOW-ALIGNED STD-PATH2))
NIL
```

A.3.3. Another Scenario in which Flow Fails

COAST is given a third scenario which is similar to the first scenario. However, in this scenario, unlike the first scenario, the pressures at the two liquids are equal. COAST explains why the amount of the liquid remains constant.

```
>(explain-observation '(:constant (ds (amount-of std-fluid5))) user:*fluid-flow-fails2-scenario*)

Explaining (CONSTANT (DS (AMOUNT-OF STD-FLUID5))) in <S-1 FLUID-FLOW-FAILS2-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW>)

  Explanation 1 for (CONSTANT (DS (AMOUNT-OF STD-FLUID5)))
    (I- (AMOUNT-OF STD-FLUID5) (A (FLUID-FLOW-RATE (FLUID-FLOW STD-FLUID5 STD-FLUID6
    STD-PATH3))))
       (INACTIVE (FLUID-FLOW STD-FLUID5 STD-FLUID6 STD-PATH3))
          (EQUAL-TO (A (PRESSURE STD-FLUID5)) (A (PRESSURE STD-FLUID6)))
NIL
```

A.4. Episode 1: Learning a New Process

COAST is given a scenario in which two solutions of different concentrations are stored in two containers separated by a partition. The layout of the scenario is shown below. The information required for experimentation is included in the facts and the transformable parameters of the description. COAST is asked to explain an observed decrease in the amount of the solution in the first container.

```
(tr-defScenario osmosis-scenario
      individuals    (solution1 solution2 vapor1 vapor2 wall partition wall-path partition-path)
      facts          ((contained-liquid solution1) (contained-liquid solution2) (contained-gas vapor1)
                     (contained-gas vapor2) (contained-fluid solution1) (contained-fluid solution2)
                     (contained-fluid vapor1) (contained-fluid vapor2) (solid wall) (solid partition)
                     (path wall-path) (path partition-path) (connection solution1 vapor1)
                     (connection solution2 vapor2) (connection wall solution1)
                     (connection wall solution2) (connection partition solution1)
                     (connection partition solution2)
                     (path-connection solution1 solution2 partition-path)
                     (path-connection solution1 solution2 wall-path) (soluble? (solute-of solution1)
                     (solvent-of solution1)) (soluble? (solute-of solution2) (solvent-of solution2))
                     (:greater-than (a (amount-of (solute-of solution1))) 0)
                     (:greater-than (a (amount-of (solute-of solution2))) 0)
                     (:greater-than (a (concentration solution1)) (a (concentration solution2)))
                     (:equal-to (a (pressure solution1)) (a (pressure solution2)))
                     ;;; information required for experimentation
                     (easily-measurable (?d (amount-of solution1)) ?scenario)
```

```
                (easily-measurable (?d (amount-of solution2)) ?scenario)
                (discriminable (?d (concentration solution1)) ?scenario ?v1 ?v2)
                (discriminable (?d (concentration solution2)) ?scenario ?v1 ?v2)
                (discriminable (?d (temperature solution1)) ?scenario ?v1 ?v2)
                (discriminable (?d (temperature solution2)) ?scenario ?v1 ?v2)
                (discriminable (?d (amount-of vapor1)) ?scenario ?v1 ?v2)
                (discriminable (?d (amount-of vapor2)) ?scenario ?v1 ?v2))
      quantities ((temperature solution1) (temperature solution2) (concentration solution1)
                (concentration solution2) (amount-of solution1) (amount-of solution2)
                (amount-of (solvent-of solution1)) (amount-of (solvent-of solution2))
                (amount-of (solute-of solution1)) (amount-of (solute-of solution2))
                (amount-of vapor1) (amount-of vapor2) (contact-area solution2 vapor2)
                (contact-area solution1 vapor1) (contact-area wall solution1)
                (contact-area wall solution2) (contact-area partition solution1)
                (contact-area partition solution2) (pressure solution1) (pressure solution2)
                (LENGTH partition-path) (LENGTH wall-path)
                (cross-sectional-area partition-path) (cross-sectional-area wall-path))
      transformable-parameters
                ;;; information required for experimentation
                ((contact-area solution1 vapor1) (contact-area solution2 vapor2)
                (contact-area wall solution1) (contact-area wall solution2)
                (contact-area partition solution1) (contact-area partition solution2)
                (cross-sectional-area partition-path) (cross-sectional-area wall-path)
                (LENGTH partition-path) (LENGTH wall-path))
      transformations nil))
>(explain-observation '(:decrease (ds (amount-of solution1))) user:*osmosis-scenario*)

Explaining (DECREASE (DS (AMOUNT-OF SOLUTION1))) in <S-1 OSMOSIS-SCENARIO> with theory:
(<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION>
<CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW>)

    Theory (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION>
    <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW>)
    failed to explain (DECREASE (DS (AMOUNT-OF SOLUTION1))) in <S-1 OSMOSIS-SCENARIO>

Explanation-based theory revision ...

    Failure type: BROKEN-EXPLANATION
    Failure subtype: UNEXPECTED-OBSERVATION
```

COAST cannot construct an explanation for the observation because all the processes that can decrease the amount of the solution are inactive due to failed conditions. Evaporation of the solution is not active because the container is closed. Absorption of the solution is not active because the two solids in contact with the solution: the wall of the cont iner and the partition, are not absorbent. Flow of solution from the container is not possible because the two paths, the partition and the wall, are not aligned for fluid flow. Therefore, COAST cannot construct an explanation for the observation and this leads to a failure.

```
Generating abstract hypotheses ...

    The hypotheses are:
```

```
((<TR-H-5 (ACTIVE? (EVAPORATION SOLUTION1 VAPOR1))
(<TR-H-4 (ACTIVE? (ABSORPTION PARTITION SOLUTION1))
(<TR-H-3 (ACTIVE? (ABSORPTION WALL SOLUTION1))
(<TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))
(<TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))
(<TR-H-6 (NEW-PROCESS? (PROCESS1797 SOLUTION1)))

Experimentation-based hypothesis refutation ...

    Designing experiments for the hypotheses:

     (<TR-H-6 (NEW-PROCESS? (PROCESS1797 SOLUTION1))>
     <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))>
     <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>
     <TR-H-3 (ACTIVE? (ABSORPTION WALL SOLUTION1))>
     <TR-H-4 (ACTIVE? (ABSORPTION PARTITION SOLUTION1))>
     <TR-H-5 (ACTIVE? (EVAPORATION SOLUTION1 VAPOR1))>)
    Scenario is: <S-1 OSMOSIS-SCENARIO>:
      Transformations: NIL

      Building ELABORATION experiment-1 for (DS (AMOUNT-OF SOLUTION2)) in
        <S-1 OSMOSIS-SCENARIO>

      Showing values supported by hypotheses:
        <PREDICTION-24 (DS (AMOUNT-OF SOLUTION2)) for <S-1 OSMOSIS-SCENARIO>
         (CONSTANT   <TR-H-5 (ACTIVE? (EVAPORATION SOLUTION1 VAPOR1))>
              <TR-H-4 (ACTIVE? (ABSORPTION PARTITION SOLUTION1))>
              <TR-H-3 (ACTIVE? (ABSORPTION WALL SOLUTION1))>)
         (INCREASE    <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
              PARTITION-PATH))>
              <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))>)

      ELABORATION Experiment:
        Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
        Scenario: <S-1 OSMOSIS-SCENARIO>

      Performing ELABORATION experiment 1 <S-1 OSMOSIS-SCENARIO>:
        (DS (AMOUNT-OF SOLUTION2)) = INCREASE

      Refuting <TR-H-5 (ACTIVE? (EVAPORATION SOLUTION1 VAPOR1))> based on
        <PREDICTION-24 (DS (AMOUNT-OF SOLUTION2)) for <S-1 OSMOSIS-SCENARIO>

      Refuting <TR-H-4 (ACTIVE? (ABSORPTION PARTITION SOLUTION1))> based on
        <PREDICTION-24 (DS (AMOUNT-OF SOLUTION2)) for <S-1 OSMOSIS-SCENARIO>

      Refuting <TR-H-3 (ACTIVE? (ABSORPTION WALL SOLUTION1))>  based on
        <PREDICTION-24 (DS (AMOUNT-OF SOLUTION2)) for <S-1 OSMOSIS-SCENARIO>
```

The first two hypotheses involve a flow of solution to the second container. Under these two hypotheses, the amount of the solution in the second container is predicted to be increasing. However, under the hypotheses involving absorption and evaporation, the amount is predicted to remain constant. COAST builds an elaboration experiment (since the amount of the solution is easy to measure) and uses the results of the experiment to refute three hypotheses.

```
    Designing experiments for the hypotheses:
```

```
(<TR-H-6 (NEW-PROCESS? (PROCESS1797 SOLUTION1))>
 <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))>
 <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>)
Scenario is: <S-2 OSMOSIS-SCENARIO>:
  Transformations: (((CROSS-SECTIONAL-AREA PARTITION-PATH) . INCREASE))

  Building ELABORATION experiment-2 for (DS (AMOUNT-OF SOLUTION1)) in <S-2
    OSMOSIS-SCENARIO>

  Building ELABORATION experiment-3 for (DS (AMOUNT-OF SOLUTION2)) in <S-2
    OSMOSIS-SCENARIO>

  Showing values supported by hypotheses:
    <PREDICTION-52 (DS (AMOUNT-OF SOLUTION2)) for <S-2 OSMOSIS-SCENARIO>
     (INCREASE   <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
          PARTITION-PATH))>
          <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
    Scenario: <S-2 OSMOSIS-SCENARIO>

  Performing ELABORATION experiment 3 <S-2 OSMOSIS-SCENARIO>:
    (DS (AMOUNT-OF SOLUTION2)) = INCREASE

  Showing values supported by hypotheses:
    <PREDICTION-53 (DS (AMOUNT-OF SOLUTION1)) for <S-2 OSMOSIS-SCENARIO>
     (DECREASE   <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
          PARTITION-PATH))>
          <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
    Scenario: <S-2 OSMOSIS-SCENARIO>

  Performing ELABORATION experiment 2 <S-2 OSMOSIS-SCENARIO>:
    (DS (AMOUNT-OF SOLUTION1)) = DECREASE

  Building ELABORATION experiment-4 for
    (DM (AMOUNT-OF SOLUTION1)) in (<S-2 OSMOSIS-SCENARIO> <S-1
    OSMOSIS-SCENARIO>)

  Showing values supported by hypotheses:
    <PREDICTION-55 (DM (AMOUNT-OF SOLUTION1)) for (<S-2 OSMOSIS-SCENARIO> <S-1
    OSMOSIS-SCENARIO>)>
     (EQUAL-TO   <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))>)
     (LESS-THAN  <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
          PARTITION-PATH))>)
     (GREATER-THAN <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
          PARTITION-PATH))>)

  Differential ELABORATION Experiment:
    Quantity to be measured: (DM (AMOUNT-OF SOLUTION1))
    New Scenario: <S-2 OSMOSIS-SCENARIO>
    Original Scenario: <S-1 OSMOSIS-SCENARIO>

  Performing ELABORATION experiment 4 (<S-2 OSMOSIS-SCENARIO> <S-1
    OSMOSIS-SCENARIO>):
    (DM (AMOUNT-OF SOLUTION1)) = GREATER-THAN

  Refuting <TR-H-1 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 WALL-PATH))> based on
    <PREDICTION-55 (DM (AMOUNT-OF SOLUTION1)) for (<S-2 OSMOSIS-SCENARIO> <S-1
    OSMOSIS-SCENARIO>)>
```

The rate of the fluid flow process is proportional to the cross-sectional area of the path (specified in the process definition). COAST uses this information to transform the original scenario into a new scenario in which the cross-sectional area of the path through the partition is increased. According to the hypothesis involving the flow of solution through the partition, the decrease in the amount of the solution in the transformed scenario is not equal to the corresponding decrease in the original scenario. According to the hypothesis involving the flow of solution through the wall, the two decreases are equal. COAST constructs a differential elaboration experiment based on these predictions and refutes the second hypothesis based on the experimental results.

```
Designing experiments for the hypotheses:

 (<TR-H-6 (NEW-PROCESS? (PROCESS1797 SOLUTION1))>
 <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>)
Scenario is: <S-3 OSMOSIS-SCENARIO>:
  Transformations: (((CROSS-SECTIONAL-AREA WALL-PATH) . INCREASE))
  Building ELABORATION experiment-5 for (DS (AMOUNT-OF SOLUTION1)) in <S-3
     OSMOSIS-SCENARIO>
  Building ELABORATION experiment-6 for (DS (AMOUNT-OF SOLUTION2)) in <S-3
     OSMOSIS-SCENARIO>
  Showing values supported by hypotheses:
     <PREDICTION-79 (DS (AMOUNT-OF SOLUTION2)) for <S-3 OSMOSIS-SCENARIO>
      (INCREASE   <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
           PARTITION-PATH))>)
  ELABORATION Experiment:
     Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
     Scenario: <S-3 OSMOSIS-SCENARIO>
  Performing ELABORATION experiment 6 <S-3 OSMOSIS-SCENARIO>:
     (DS (AMOUNT-OF SOLUTION2)) = INCREASE
  Showing values supported by hypotheses:
     <PREDICTION-80 (DS (AMOUNT-OF SOLUTION1)) for <S-3 OSMOSIS-SCENARIO>
      (DECREASE <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
           PARTITION-PATH))>)
  ELABORATION Experiment:
     Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
     Scenario: <S-3 OSMOSIS-SCENARIO>
  Performing ELABORATION experiment 5 <S-3 OSMOSIS-SCENARIO>:
     (DS (AMOUNT-OF SOLUTION1)) = DECREASE
  Building ELABORATION experiment-7 for (DM (AMOUNT-OF SOLUTION1)) in (<S-3
     OSMOSIS-SCENARIO> <S-1 OSMOSIS-SCENARIO>)
  Showing values supported by hypotheses:
     <PREDICTION-81 (DM (AMOUNT-OF SOLUTION1)) for (<S-3 OSMOSIS-SCENARIO> <S-1
     OSMOSIS-SCENARIO>)>
      (EQUAL-TO   <TR-H-2 (ACTIVE? (FLUID-FLOW SOLUTION1 SOLUTION2
           PARTITION-PATH))>)
  Differential ELABORATION Experiment:
     Quantity to be measured: (DM (AMOUNT-OF SOLUTION1))
```

New Scenario: <S-3 OSMOSIS-SCENARIO>
Original Scenario: <S-1 OSMOSIS-SCENARIO>

Performing ELABORATION experiment 7 (<S-3 OSMOSIS-SCENARIO> <S-1 OSMOSIS-SCENARIO>):
(DM (AMOUNT-OF SOLUTION1)) = EQUAL-TO

Exemplar-based theory rejection ...

Refining hypotheses ...

The hypotheses are:

((<TR-H-12 (NEW-PROCESS (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)
(<TR-H-10 (NEGATE-CONDITION (LESS-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))) (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))))>
ACTIVE?-CONDITIONS?)
(<TR-H-7 (RETRACT-CONDITION (FLUID-FLOW-ALIGNED PARTITION-PATH) (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))> ACTIVE?-CONDITIONS?)
(<TR-H-11 (NEGATE-CONDITION (EQUAL-TO (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))) (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))))>
ACTIVE?-CONDITIONS?)
(<TR-H-9 (RETRACT-CONDITION (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))) (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>
ACTIVE?-CONDITIONS?)
(<TR-H-8 (NEGATE-CONDITION (NOT (FLUID-FLOW-ALIGNED PARTITION-PATH))
(FLUID-FLOW-ALIGNED PARTITION-PATH))> ACTIVE?-CONDITIONS?))

Experimentation-based hypothesis refutation ...

Exemplar-based theory rejection ...

COAST constructs revised theories based on each refined hypothesis and employs exemplar–based theory rejection to test each of these theories. If all the revised theories corresponding to a hypothesis are rejected then the hypothesis is eliminated.

Verifying consistency with exemplars

Hypothesis: <TR-H-10 (NEGATE-CONDITION (LESS-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))) (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE SOLUTION2))))>

Exemplars retrieved are:
(<EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1)))
<S-1 FLUID-FLOW-WORKS-SCENARIO>>
<EXEMPLAR-3 (CONSTANT (DS (AMOUNT-OF STD-FLUID5)))
<S-1 FLUID-FLOW-FAILS2-SCENARIO>>)

Checking exemplar: <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1 FLUID-FLOW-WORKS-SCENARIO>>

Cwas under hypothesis: (<CWA-13 FLUID-FLOW> <CWA-10 FLUID-FLOW>)

Refuting <CWA-13 FLUID-FLOW> as <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1 FLUID-FLOW-WORKS-SCENARIO>> cannot be explained

```
Refuting <CWA-10 FLUID-FLOW> as <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF
   STD-FLUID1))) <S-1 FLUID-FLOW-WORKS-SCENARIO>> cannot be explained

Exemplar-based rejection of hypothesis:
   <TR-H-10 (NEGATE-CONDITION (LESS-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE
   SOLUTION2))) (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE
   SOLUTION2))))>

Verifying consistency with exemplars


Hypothesis: <TR-H-7 (RETRACT-CONDITION (FLUID-FLOW-ALIGNED PARTITION-PATH)
   (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>

Exemplars retrieved are:
   (<EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
   FLUID-FLOW-WORKS-SCENARIO>>
   <EXEMPLAR-2 (CONSTANT (DS (AMOUNT-OF STD-FLUID3))) <S-1
   FLUID-FLOW-FAILS1-SCENARIO>>)

Checking exemplar: <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
   FLUID-FLOW-WORKS-SCENARIO>>

Cwas under hypothesis: (<CWA-12 FLUID-FLOW> <CWA-11 FLUID-FLOW>)

Refuting <CWA-12 FLUID-FLOW> as <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF
   STD-FLUID1))) <S-1 FLUID-FLOW-WORKS-SCENARIO>> cannot be explained

Re-explaining exemplar under cwa <CWA-11 FLUID-FLOW>

Explanation 308 for (DECREASE (DS (AMOUNT-OF STD-FLUID1)))
  (I- (AMOUNT-OF STD-FLUID1) (A (FLUID-FLOW-RATE (FLUID-FLOW STD-FLUID1 STD-FLUID2
  STD-PATH1))))
     (ACTIVE (FLUID-FLOW STD-FLUID1 STD-FLUID2 STD-PATH1))
```

The revised theory involves the retraction of both the conditions of the flow process. Therefore, the flow process is active in the given scenario and COAST can explain the exemplar observation. However, the same theory cannot explain the other exemplars (see below) and is therefore rejected.

```
Checking exemplar: <EXEMPLAR-2 (CONSTANT (DS (AMOUNT-OF STD-FLUID3))) <S-1
   FLUID-FLOW-FAILS1-SCENARIO>>

Cwas under hypothesis: (<CWA-11 FLUID-FLOW>)

Refuting <CWA-11 FLUID-FLOW> as <EXEMPLAR-2 (CONSTANT (DS (AMOUNT-OF
   STD-FLUID3))) <S-1 FLUID-FLOW-FAILS1-SCENARIO>> cannot be explained

Exemplar-based rejection of hypothesis:
   <TR-H-7 (RETRACT-CONDITION (FLUID-FLOW-ALIGNED PARTITION-PATH) (FLUID-FLOW
   SOLUTION1 SOLUTION2 PARTITION-PATH))>

Verifying consistency with exemplars


Hypothesis: <TR-H-11 (NEGATE-CONDITION (EQUAL-TO (A (PRESSURE SOLUTION1)) (A
   (PRESSURE SOLUTION2))) (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE
   SOLUTION2))))>

Exemplars retrieved are:
   (<EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
   FLUID-FLOW-WORKS-SCENARIO>>
```

<EXEMPLAR-3 (CONSTANT (DS (AMOUNT-OF STD-FLUID5))) <S-1
FLUID-FLOW-FAILS2-SCENARIO>>)

Checking exemplar: <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
FLUID-FLOW-WORKS-SCENARIO>>

Cwas under hypothesis: (<CWA-9 FLUID-FLOW>)

Refuting <CWA-9 FLUID-FLOW> as <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF
STD-FLUID1))) <S-1 FLUID-FLOW-WORKS-SCENARIO>> cannot be explained

Exemplar-based rejection of hypothesis:
<TR-H-11 (NEGATE-CONDITION (EQUAL-TO (A (PRESSURE SOLUTION1)) (A (PRESSURE
SOLUTION2))) (GREATER-THAN (A (PRESSURE SOLUTION1)) (A (PRESSURE
SOLUTION2))))>

Verifying consistency with exemplars

Hypothesis: <TR-H-9 (RETRACT-CONDITION (GREATER-THAN (A (PRESSURE SOLUTION1)) (A
(PRESSURE SOLUTION2))) (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>

Exemplars retrieved are:
(<EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
FLUID-FLOW-WORKS-SCENARIO>>
<EXEMPLAR-3 (CONSTANT (DS (AMOUNT-OF STD-FLUID5))) <S-1
FLUID-FLOW-FAILS2-SCENARIO>>)

Checking exemplar: <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
FLUID-FLOW-WORKS-SCENARIO>>

Cwas under hypothesis: (<CWA-8 FLUID-FLOW>)

Refuting <CWA-8 FLUID-FLOW> as <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF
STD-FLUID1))) <S-1 FLUID-FLOW-WORKS-SCENARIO>> cannot be explained

Exemplar-based rejection of hypothesis:
<TR-H-9 (RETRACT-CONDITION (GREATER-THAN (A (PRESSURE SOLUTION1)) (A
(PRESSURE SOLUTION2))) (FLUID-FLOW SOLUTION1 SOLUTION2 PARTITION-PATH))>

Verifying consistency with exemplars

Hypothesis: <TR-H-8 (NEGATE-CONDITION (NOT (FLUID-FLOW-ALIGNED PARTITION-PATH))
(FLUID-FLOW-ALIGNED PARTITION-PATH))>

Exemplars retrieved are:
(<EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
FLUID-FLOW-WORKS-SCENARIO>>
<EXEMPLAR-2 (CONSTANT (DS (AMOUNT-OF STD-FLUID3))) <S-1
FLUID-FLOW-FAILS1-SCENARIO>>)

Checking exemplar: <EXEMPLAR-1 (DECREASE (DS (AMOUNT-OF STD-FLUID1))) <S-1
FLUID-FLOW-WORKS-SCENARIO>>

Cwas under hypothesis: NIL

Exemplar-based rejection of hypothesis:
<TR-H-8 (NEGATE-CONDITION (NOT (FLUID-FLOW-ALIGNED PARTITION-PATH))
(FLUID-FLOW-ALIGNED PARTITION-PATH))>

Refining hypotheses ...

No more refinement ...

Only one theory, corresponding to the inclusion of a new process, remains after experimentation–based hypothesis refutation and exemplar–based theory rejection. COAST can construct an explanation for the observation using this theory.

```
Explaining (DECREASE (DS (AMOUNT-OF SOLUTION1))) in <S-1 OSMOSIS-SCENARIO> with theory:
(<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION>
<CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14 PROCESS1797>)

  Explanation 334 for (DECREASE (DS (AMOUNT-OF SOLUTION1)))
    (I- (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
    PARTITION-PATH))))
       (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
          (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)

  Final theory is:
     ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14
     PROCESS1797>))

Theory revision completed

NIL

>(process-definitions-from-name 'process1797)

   ((DEFPROCESS (PROCESS1797 ?VAR1807 ?VAR1808 ?VAR1809)
       INDIVIDUALS              ((?VAR1807 (CONTAINED-FLUID ?VAR1807)
                                           (CONTAINED-LIQUID?VAR1807))
                                 (?VAR1808 (CONTAINED-FLUID ?VAR1808)
                                           (CONTAINED-LIQUID ?VAR1808))
                                 (?VAR1809 (PATH ?VAR1809)))
       PRECONDITIONS            ((PRECONDITION1806 ?VAR1807 ?VAR1808 ?VAR1809))
       QUANTITYCONDITIONS       NIL
       RELATIONS                ((Q+ (PROCESS1797-RATE ?SELF)
                                     (CROSS-SECTIONAL-AREA ?VAR1809)))
       INFLUENCES               ((I+ (AMOUNT-OF ?VAR1808) (A (PROCESS1797-RATE ?SELF)))
                                 (I- (AMOUNT-OF ?VAR1807) (A (PROCESS1797-RATE ?SELF))))))

>(setf permeable 'precondition1806)

PRECONDITION1806
```

The new process incorporates the information obtained during experimentation: the amount of solution2 decreases and the rate of the process depends on the cross–sectional area of the path. Therefore, the new theory can construct explanations for these observations also.

```
>(explain-observation '(:increase (ds (amount-of solution2))) user:*osmosis-scenario*)

Explaining (INCREASE (DS (AMOUNT-OF SOLUTION2))) in <S-1 OSMOSIS-SCENARIO> with theory:
(<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION>
<CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14 PROCESS1797>)
```

```
Explanation 335 for (INCREASE (DS (AMOUNT-OF SOLUTION2)))
  (I+ (AMOUNT-OF SOLUTION2) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
  PARTITION-PATH))))
     (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
        (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)
NIL
```

A.5. Episode 2: Correcting an Influence

COAST is asked to explain an observed increase in the concentration of the first solution in a scenario similar to one described earlier. Due to space limitations only important portions of the trace are shown.

```
>(explain-observation '(:increase (ds (concentration solution1))) (EVAL *osmosis-conc-scenario*))

Explaining (INCREASE (DS (CONCENTRATION SOLUTION1))) in <S-1 OSMOSIS-CONC-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14
PROCESS1797>)

    Theory (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
    ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14
    PROCESS1797>) failed to explain (INCREASE (DS (CONCENTRATION SOLUTION1))) in <S-1
    OSMOSIS-CONC-SCENARIO>
```

COAST cannot construct an explanation because the only active process, an instance of the new process, Process1797, does not affect the concentration of the first solution since it involves the flow of the solution as a whole.

```
Explanation-based theory revision ...

    Failure type: BROKEN-EXPLANATION
    Failure subtype: UNEXPECTED-OBSERVATION

Generating abstract hypotheses ...

    The hypotheses are:

     ((<TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH) (INCREASE
     (DS (CONCENTRATION SOLUTION1))))> CAUSES?)
     (<TR-H-2 (NEW-PROCESS? (PROCESS1822 SOLUTION1))> NEW-PROCESS?))

Experimentation-based hypothesis refutation ...

    Designing experiments for the hypotheses:

     (<TR-H-2 (NEW-PROCESS? (PROCESS1822 SOLUTION1))>
     <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH) (INCREASE (DS
     (CONCENTRATION SOLUTION1))))>)
```

```
Scenario is:
  <S-1 OSMOSIS-CONC-SCENARIO>:
  Transformations: NIL

  Building ELABORATION experiment-1 for (DS (AMOUNT-OF SOLUTION1)) in <S-1
    OSMOSIS-CONC-SCENARIO>

  Building ELABORATION experiment-2 for (DS (AMOUNT-OF SOLUTION2)) in <S-1
    OSMOSIS-CONC-SCENARIO>

  Showing values supported by hypotheses:
    <PREDICTION-33 (DS (AMOUNT-OF SOLUTION2)) for <S-1 OSMOSIS-CONC-SCENARIO>
      (INCREASE  <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2
          PARTITION-PATH) (INCREASE (DS (CONCENTRATION SOLUTION1)))))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
    Scenario: <S-1 OSMOSIS-CONC-SCENARIO>

  Performing ELABORATION experiment 2 <S-1 OSMOSIS-CONC-SCENARIO>:
    (DS (AMOUNT-OF SOLUTION2)) = INCREASE

  Showing values supported by hypotheses:
    <PREDICTION-34 (DS (AMOUNT-OF SOLUTION1)) for <S-1 OSMOSIS-CONC-SCENARIO>
     (DECREASE  <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2
          PARTITION-PATH) (INCREASE (DS (CONCENTRATION SOLUTION1)))))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
    Scenario: <S-1 OSMOSIS-CONC-SCENARIO>

  Performing ELABORATION experiment 1 <S-1 OSMOSIS-CONC-SCENARIO>:
    (DS (AMOUNT-OF SOLUTION1)) = DECREASE

Designing experiments for the hypotheses:


 (<TR-H-2 (NEW-PROCESS? (PROCESS1822 SOLUTION1))>
 <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH) (INCREASE (DS
 (CONCENTRATION SOLUTION1)))))>)


Scenario is:
  <S-2 OSMOSIS-CONC-SCENARIO>:
  Transformations: (((PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH) . FALSE))

  Building ELABORATION experiment-3 for (DS (AMOUNT-OF SOLUTION1)) in <S-2
    OSMOSIS-CONC-SCENARIO>

  Building ELABORATION experiment-4 for (DS (AMOUNT-OF SOLUTION2)) in <S-2
    OSMOSIS-CONC-SCENARIO>

  Building ELABORATION experiment-5 for (DS (CONCENTRATION SOLUTION1)) in <S-2
    OSMOSIS-CONC-SCENARIO>

  Showing values supported by hypotheses:
    <PREDICTION-56 (DS (CONCENTRATION SOLUTION1)) for <S-2
    OSMOSIS-CONC-SCENARIO>
     (CONSTANT  <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2
          PARTITION-PATH) (INCREASE (DS (CONCENTRATION SOLUTION1)))))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (CONCENTRATION SOLUTION1))
    Scenario: <S-2 OSMOSIS-CONC-SCENARIO>
```

Performing ELABORATION experiment 5 <S-2 OSMOSIS-CONC-SCENARIO>:
(DS (CONCENTRATION SOLUTION1)) = CONSTANT

Showing values supported by hypotheses:
<PREDICTION-65 (DS (AMOUNT-OF SOLUTION2)) for <S-2 OSMOSIS-CONC-SCENARIO>
(CONSTANT <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2
PARTITION-PATH) (INCREASE (DS (CONCENTRATION SOLUTION1)))))>)

ELABORATION Experiment:
Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
Scenario: <S-2 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 4 <S-2 OSMOSIS-CONC-SCENARIO>:
(DS (AMOUNT-OF SOLUTION2)) = CONSTANT

Showing values supported by hypotheses:
<PREDICTION-66 (DS (AMOUNT-OF SOLUTION1)) for <S-2 OSMOSIS-CONC-SCENARIO>
(CONSTANT <TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2
PARTITION-PATH) (INCREASE (DS (CONCENTRATION SOLUTION1)))))>)

ELABORATION Experiment:
Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
Scenario: <S-2 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 3 <S-2 OSMOSIS-CONC-SCENARIO>:
(DS (AMOUNT-OF SOLUTION1)) = CONSTANT

Exemplar-based theory rejection ...

Refining hypotheses ...

The hypotheses are:

((<TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>)
(<TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> CAUSES?-EFFECTS?)
(<TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION2))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> CAUSES?-EFFECTS?)
(<TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
(PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
(AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
PARTITION-PATH))))))> CAUSES?-EFFECTS?)
(<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
(PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> CAUSES?-EFFECTS?))

Experimentation-based hypothesis refutation ...

Designing experiments for the hypotheses:

(<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
(PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
(PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
(AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
PARTITION-PATH))))))>
<TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION2))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1))

```
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>)
Scenario is:
  <S-4 OSMOSIS-CONC-SCENARIO>:
  Transformations: (((FLUID-FLOW-ALIGNED SOLUTION2-PATH) . TRUE))
```

COAST constructs a new scenario in which a path connecting solution2 to another solution is opened leading to a flow of solution out of the second container.

```
Building ELABORATION experiment-6 for (DS (AMOUNT-OF SOLUTION1)) in <S-4
  OSMOSIS-CONC-SCENARIO>

Building ELABORATION experiment-7 for (DS (AMOUNT-OF SOLUTION2)) in <S-4
  OSMOSIS-CONC-SCENARIO>

Building ELABORATION experiment-8 for (DS (CONCENTRATION SOLUTION1)) in <S-4
  OSMOSIS-CONC-SCENARIO>

Showing values supported by hypotheses:
  <PREDICTION-100 (DS (CONCENTRATION SOLUTION1)) for <S-4
  OSMOSIS-CONC-SCENARIO>
   (INCREASE   <TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>
       <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF
       SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
       <TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF
       SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
       <TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
       (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1
       SOLUTION2 PARTITION-PATH)))))>
       <TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
       (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
  Quantity to be measured: (DS (CONCENTRATION SOLUTION1))
  Scenario: <S-4 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 8 <S-4 OSMOSIS-CONC-SCENARIO>:
  (DS (CONCENTRATION SOLUTION1)) = INCREASE

Showing values supported by hypotheses:
  <PREDICTION-103 (DS (AMOUNT-OF SOLUTION2)) for <S-4 OSMOSIS-CONC-SCENARIO>
   (UNKNOWN   <TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>
       <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF
       SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
       <TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF
       SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
       <TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
       (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1
       SOLUTION2 PARTITION-PATH)))))>
       <TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
       (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
  Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
  Scenario: <S-4 OSMOSIS-CONC-SCENARIO>
```

```
Performing ELABORATION experiment 7 <S-4 OSMOSIS-CONC-SCENARIO>:
  (DS (AMOUNT-OF SOLUTION2)) = INCREASE

Showing values supported by hypotheses:
  <PREDICTION-104 (DS (AMOUNT-OF SOLUTION1)) for <S-4 OSMOSIS-CONC-SCENARIO>
   (DECREASE  <TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>
       <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF
       SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
       <TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF
       SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
       <TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
       (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1
       SOLUTION2 PARTITION-PATH)))))>
       <TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
       (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
  Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
  Scenario: <S-4 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 6 <S-4 OSMOSIS-CONC-SCENARIO>:
  (DS (AMOUNT-OF SOLUTION1)) = DECREASE

Building ELABORATION experiment-9 for (DM (CONCENTRATION SOLUTION1)) in (<S-4
  OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>)

Showing values supported by hypotheses:
  <PREDICTION-106 (DM (CONCENTRATION SOLUTION1)) for (<S-4
  OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>)>
   (EQUAL-TO   <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1)
       (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2
       PARTITION-PATH))>
       <TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
       (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1
       SOLUTION2 PARTITION-PATH)))))>
       <TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
       (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
       (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)
   (LESS-THAN  <TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1)
       (AMOUNT-OF SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2
       PARTITION-PATH))>)
   (GREATER-THAN <TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1)
       (AMOUNT-OF SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2
       PARTITION-PATH))>)
```

The hypothesis that links the concentration of *solution1* with the *amount of solution2* predicts that the change in the concentration of solution1 is not the same as the change in the concentration of solution1 in the original scenario since the change in the amount of solution2 is different in the two scenarios. The other hypotheses predict that the change in concentration is the same in both scenarios. COAST uses this prediction to construct a differential elaboration experiment.

```
Differential ELABORATION Experiment:
  Quantity to be measured: (DM (CONCENTRATION SOLUTION1))
```

New Scenario: <S-4 OSMOSIS-CONC-SCENARIO>
Original Scenario: <S-1 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 9 (<S-4 OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>):
(DM (CONCENTRATION SOLUTION1)) = EQUAL-TO

Refuting <TR-H-5 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> based on <PREDICTION-106 (DM (CONCENTRATION SOLUTION1)) for (<S-4 OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>)>

Designing experiments for the hypotheses:

(<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I- (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))))>
<TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>)

Scenario is:
<S-3 OSMOSIS-CONC-SCENARIO>:
Transformations: (((FLUID-FLOW-ALIGNED SOLUTION1-PATH) . TRUE))

Building ELABORATION experiment-10 for (DS (AMOUNT-OF SOLUTION1)) in <S-3 OSMOSIS-CONC-SCENARIO>

Building ELABORATION experiment-11 for (DS (AMOUNT-OF SOLUTION2)) in <S-3 OSMOSIS-CONC-SCENARIO>

Building ELABORATION experiment-12 for (DS (CONCENTRATION SOLUTION1)) in <S-3 OSMOSIS-CONC-SCENARIO>

Showing values supported by hypotheses: <PREDICTION-138 (DS (CONCENTRATION SOLUTION1)) for <S-3 OSMOSIS-CONC-SCENARIO>
(INCREASE <TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>
<TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I- (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))))>
<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
Quantity to be measured: (DS (CONCENTRATION SOLUTION1))
Scenario: <S-3 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 12 <S-3 OSMOSIS-CONC-SCENARIO>:
(DS (CONCENTRATION SOLUTION1)) = INCREASE

Showing values supported by hypotheses:
<PREDICTION-141 (DS (AMOUNT-OF SOLUTION2)) for <S-3 OSMOSIS-CONC-SCENARIO>
(INCREASE <TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>
<TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-

(AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))))>
<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
Scenario: <S-3 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 11 <S-3 OSMOSIS-CONC-SCENARIO>:
(DS (AMOUNT-OF SOLUTION2)) = INCREASE

Showing values supported by hypotheses:
<PREDICTION-142 (DS (AMOUNT-OF SOLUTION1)) for <S-3 OSMOSIS-CONC-SCENARIO>
(DECREASE <TR-H-7 (NEW-PROCESS (PROCESS1822 SOLUTION2 SOLUTION1))>
<TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I- (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))))>
<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
Scenario: <S-3 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 10 <S-3 OSMOSIS-CONC-SCENARIO>:
(DS (AMOUNT-OF SOLUTION1)) = DECREASE

Building ELABORATION experiment-13 for (DM (CONCENTRATION SOLUTION1)) in (<S-3 OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>)

Showing values supported by hypotheses:
<PREDICTION-144 (DM (CONCENTRATION SOLUTION1)) for (<S-3 OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>)>
(EQUAL-TO <TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I- (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))))>
<TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)
(LESS-THAN <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)
(GREATER-THAN <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>)

Differential ELABORATION Experiment:
Quantity to be measured: (DM (CONCENTRATION SOLUTION1))
New Scenario: <S-3 OSMOSIS-CONC-SCENARIO>
Original Scenario: <S-1 OSMOSIS-CONC-SCENARIO>

Performing ELABORATION experiment 13 (<S-3 OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>):
(DM (CONCENTRATION SOLUTION1)) = EQUAL-TO

Refuting <TR-H-6 (NEW-RELATION (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF SOLUTION1)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> based on

```
<PREDICTION-144 (DM (CONCENTRATION SOLUTION1)) for (<S-3
OSMOSIS-CONC-SCENARIO> <S-1 OSMOSIS-CONC-SCENARIO>)>

Exemplar-based theory rejection ...

  Verifying consistency with exemplars

    Hypothesis: <TR-H-3 (MODIFIED-INFLUENCE (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A
      (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I-
      (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
      PARTITION-PATH))))))>

    Exemplars retrieved are:
      (<EXEMPLAR-5 (DECREASE (DS (AMOUNT-OF SOLUTION1))) <S-1 OSMOSIS-SCENARIO>>)

    Checking exemplar: <EXEMPLAR-5 (DECREASE (DS (AMOUNT-OF SOLUTION1))) <S-1
      OSMOSIS-SCENARIO>>

    Cwas under hypothesis: (<CWA-15 PROCESS1797>)

    Re-explaining exemplar under cwa <CWA-15 PROCESS1797>

Explanation 682 for (DECREASE (DS (AMOUNT-OF SOLUTION1)))
  (Q+ (AMOUNT-OF SOLUTION1) (AMOUNT-OF (SOLVENT-OF SOLUTION1)))
     (ACTIVE (SOLUTION SOLUTION1))
        (GREATER-THAN (A (AMOUNT-OF (SOLUTE-OF SOLUTION1))) 0)
        (SOLUBLE? (SOLUTE-OF SOLUTION1) (SOLVENT-OF SOLUTION1))
  (DECREASE (DS (AMOUNT-OF (SOLVENT-OF SOLUTION1))))
     (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797
     SOLUTION1 SOLUTION2 PARTITION-PATH)))) (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2
     PARTITION-PATH))
        (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)
```

Notice that the explanation for this exemplar observation is different from the explanation constructed using the original theory. The original theory explained the observation as a direct influence of the process.

```
  Verifying consistency with exemplars

    Hypothesis: <TR-H-4 (NEW-INFLUENCE (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A
      (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
      (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>

    Exemplars retrieved are:
      (<EXEMPLAR-6 (INCREASE (DS (AMOUNT-OF SOLUTION2))) <S-1 OSMOSIS-SCENARIO>>
      <EXEMPLAR-5 (DECREASE (DS (AMOUNT-OF SOLUTION1))) <S-1 OSMOSIS-SCENARIO>>)

    Checking exemplar: <EXEMPLAR-6 (INCREASE (DS (AMOUNT-OF SOLUTION2))) <S-1
      OSMOSIS-SCENARIO>>

    Cwas under hypothesis: (<CWA-16 PROCESS1797>)

    Re-explaining exemplar under cwa <CWA-16 PROCESS1797>
```

```
Explanation 710 for (INCREASE (DS (AMOUNT-OF SOLUTION2)))
  (I+ (AMOUNT-OF SOLUTION2) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
  PARTITION-PATH))))
     (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
        (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)

     Checking exemplar: <EXEMPLAR-5 (DECREASE (DS (AMOUNT-OF SOLUTION1))) <S-1
        OSMOSIS-SCENARIO>>

     Cwas under hypothesis: (<CWA-16 PROCESS1797>)

     Re-explaining exemplar under cwa <CWA-16 PROCESS1797>

Explanation 709 for (DECREASE (DS (AMOUNT-OF SOLUTION1)))
  (I- (AMOUNT-OF SOLUTION1) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2
  PARTITION-PATH))))
     (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
        (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)

Refining hypotheses ...

   No more refinement ...
```

Three competing theories remain after experimentation–based hypothesis refutation and exemplar–based theory rejection: a theory in which an influence of the new process, Process1797, is revised, a theory in which a new influence is added to the new process, Process1797, and a theory in which a new process, Process1822, is added[1]. COAST can construct an explanation for the observation that led to the failure using each of these theories.

```
Explaining (INCREASE (DS (CONCENTRATION SOLUTION1))) in <S-1 OSMOSIS-CONC-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
PROCESS1797>)

  Explanation 741 for (INCREASE (DS (CONCENTRATION SOLUTION1)))
    (Q- (CONCENTRATION SOLUTION1) (AMOUNT-OF (SOLVENT-OF SOLUTION1)))
       (ACTIVE (SOLUTION SOLUTION1))
          (GREATER-THAN (A (AMOUNT-OF (SOLUTE-OF SOLUTION1))) 0)
          (SOLUBLE? (SOLUTE-OF SOLUTION1) (SOLVENT-OF SOLUTION1))
    (DECREASE (DS (AMOUNT-OF (SOLVENT-OF SOLUTION1))))
       (I- (AMOUNT-OF (SOLVENT-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797
       SOLUTION1 SOLUTION2 PARTITION-PATH))))
          (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
             (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)

Explaining (INCREASE (DS (CONCENTRATION SOLUTION1))) in <S-1 OSMOSIS-CONC-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-16
PROCESS1797>)
```

[1] COAST cannot construct an experiment to discriminate between the theory incorporating the modified influence and the theory incorporating the new influence even though they make different predictions about the amount of the solute in the solution because the initial scenario description specifies that the amount of solute in the solution cannot be experimentally measured.

```
Explanation 810 for (INCREASE (DS (CONCENTRATION SOLUTION1)))
  (Q+ (CONCENTRATION SOLUTION1) (AMOUNT-OF (SOLUTE-OF SOLUTION1)))
     (ACTIVE (SOLUTION SOLUTION1))
        (GREATER-THAN (A (AMOUNT-OF (SOLUTE-OF SOLUTION1))) 0)
        (SOLUBLE? (SOLUTE-OF SOLUTION1) (SOLVENT-OF SOLUTION1))
  (INCREASE (DS (AMOUNT-OF (SOLUTE-OF SOLUTION1))))
     (I+ (AMOUNT-OF (SOLUTE-OF SOLUTION1)) (A (PROCESS1797-RATE (PROCESS1797
     SOLUTION1 SOLUTION2 PARTITION-PATH))))
        (ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
           (PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)

Explaining (INCREASE (DS (CONCENTRATION SOLUTION1))) in <S-1 OSMOSIS-CONC-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14
PROCESS1797> <CWA-19 PROCESS1822>)

Explanation 887 for (INCREASE (DS (CONCENTRATION SOLUTION1)))
  (I+ (CONCENTRATION SOLUTION1) (A (PROCESS1822-RATE (PROCESS1822 SOLUTION2
  SOLUTION1))))
     (ACTIVE (PROCESS1822 SOLUTION2 SOLUTION1))
        (PRECONDITION1823 SOLUTION2 SOLUTION1)

  Selecting best theory ...


Theories are:

     ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
     PROCESS1797>)

     (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-16
     PROCESS1797>)

     (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14
     PROCESS1797> <CWA-19 PROCESS1822>))


Rating for each theory based on structural complexity is

     (((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
     PROCESS1797>) 59)

     ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-16
     PROCESS1797>) 60)

     ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14
     PROCESS1797> <CWA-19 PROCESS1822>) 66))


Rating for each theory based on explanations is

     (((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
     ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
     PROCESS1797>) 24)
```

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-16 PROCESS1797>) 17)

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14 PROCESS1797> <CWA-19 PROCESS1822>) 10))

The theory that includes the new process provides the simplest explanations because it explains the observation as a direct effect of the new process. The theory that includes the modified influence constructs longer explanations than the other theories for some observations (such as the observed decrease in the amount of solution1 in the exemplar).

Rating for each theory based on predictive power is

(((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15 PROCESS1797>) 10)

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-16 PROCESS1797>) 10)

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14 PROCESS1797> <CWA-19 PROCESS1822>) 7))

Rating for each theory is

(((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15 PROCESS1797>) 1211/330)

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-16 PROCESS1797>) 4867/1320)

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-14 PROCESS1797> <CWA-19 PROCESS1822>) 263/60))

After normalizing and weighting the contributions from each criterion, COAST finds the best theory to be the theory with the modified influence.

Final theory is:

((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15 PROCESS1797>))

Theory revision completed

```
NIL

>(process-definitions-from-name 'process1797)

  (DEFPROCESS (PROCESS1797 ?VAR1807 ?VAR1808 ?VAR1809)
     INDIVIDUALS            ((?VAR1807 (CONTAINED-FLUID ?VAR1807)
                                   (CONTAINED-LIQUID ?VAR1807))
                             (?VAR1808 (CONTAINED-FLUID ?VAR1808)
                                   (CONTAINED-LIQUID ?VAR1808))
                             (?VAR1809 (PATH ?VAR1809)))
     PRECONDITIONS          ((PRECONDITION1806 ?VAR1807 ?VAR1808 ?VAR1809))
     QUANTITYCONDITIONS     NIL
     RELATIONS              ((Q+ (PROCESS1797-RATE ?SELF)
                                   (CROSS-SECTIONAL-AREA ?VAR1809)))
     INFLUENCES             ((I+ (AMOUNT-OF ?VAR1808) (A (PROCESS1797-RATE ?SELF)))
                             (I- (AMOUNT-OF (SOLVENT-OF ?VAR1807))
                                   (A (PROCESS1797-RATE ?SELF))))))
```

A.6. Episode 3: Correcting Another Influence

COAST is asked to explain an observed decrease in the concentration of solution2 in a scenario similar to the previous scenario. This episode is similar in many respects to the previous one and therefore a very brief trace is provided.

```
>(explain-observation '(:decrease (ds (concentration solution2))) (EVAL *osmosis-conc2-scenario*))

Explaining (DECREASE (DS (CONCENTRATION SOLUTION2))) in <S-1 OSMOSIS-CONC2-SCENARIO>
with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
PROCESS1797>)

    Theory (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
    ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
    PROCESS1797>) failed to explain (DECREASE (DS (CONCENTRATION SOLUTION2))) in <S-1
    OSMOSIS-CONC2-SCENARIO>

Explanation-based theory revision ...

    Failure type: BROKEN-EXPLANATION
    Failure subtype: UNEXPECTED-OBSERVATION

Generating abstract hypotheses ...

    The hypotheses are:

    ((<TR-H-1 (CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH) (DECREASE
    (DS (CONCENTRATION SOLUTION2))))> CAUSES?)
    (<TR-H-2 (NEW-PROCESS? (PROCESS1838 SOLUTION2))> NEW-PROCESS?))

Experimentation-based hypothesis refutation ...

Exemplar-based theory rejection ...
```

Refining hypotheses ...

The hypotheses are:

**((<TR-H-7 (NEW-PROCESS (PROCESS1838 SOLUTION1 SOLUTION2))>)
(<TR-H-6 (NEW-RELATION (Q+ (CONCENTRATION SOLUTION2) (AMOUNT-OF (SOLVENT-OF SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> CAUSES?-EFFECTS?)
(<TR-H-5 (NEW-RELATION (Q- (CONCENTRATION SOLUTION2) (AMOUNT-OF SOLUTION2)) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> CAUSES?-EFFECTS?)
(<TR-H-3 (MODIFIED-INFLUENCE (I+ (AMOUNT-OF (SOLVENT-OF SOLUTION2)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH)))) (I+ (AMOUNT-OF SOLUTION2) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))))> CAUSES?-EFFECTS?)
(<TR-H-4 (NEW-INFLUENCE (I- (AMOUNT-OF (SOLUTE-OF SOLUTION2)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> CAUSES?-EFFECTS?))**

Experimentation-based hypothesis refutation ...

Exemplar-based theory rejection ...

No more refinement ...

In this episode, the amount of solute and solvent of the solutions are specified to be experimentally measurable in the given scenario. Therefore, experimentation-based hypothesis refutation is able to discriminate between the modified influence and the new influence. Two competing theories remain after experimentation-based hypothesis refutation and exemplar-based theory rejection.

Explaining (DECREASE (DS (CONCENTRATION SOLUTION2))) in <S-1 OSMOSIS-CONC2-SCENARIO> with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-20 PROCESS1797>)

**Explanation 1457 for (DECREASE (DS (CONCENTRATION SOLUTION2)))
(Q- (CONCENTRATION SOLUTION2) (AMOUNT-OF (SOLVENT-OF SOLUTION2)))
(ACTIVE (SOLUTION SOLUTION2))
(GREATER-THAN (A (AMOUNT-OF (SOLUTE-OF SOLUTION2))) 0)
(SOLUBLE? (SOLUTE-OF SOLUTION2) (SOLVENT-OF SOLUTION2)) (INCREASE (DS (AMOUNT-OF (SOLVENT-OF SOLUTION2))))
(I+ (AMOUNT-OF (SOLVENT-OF SOLUTION2)) (A (PROCESS1797-RATE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))))
(ACTIVE (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))
(PRECONDITION1806 SOLUTION1 SOLUTION2 PARTITION-PATH)**

Explaining (DECREASE (DS (CONCENTRATION SOLUTION2))) in <S-1 OSMOSIS-CONC2-SCENARIO> with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15 PROCESS1797> <CWA-24 PROCESS1838>)

```
Explanation 1544 for (DECREASE (DS (CONCENTRATION SOLUTION2)))
  (I- (CONCENTRATION SOLUTION2) (A (PROCESS1838-RATE (PROCESS1838 SOLUTION1
  SOLUTION2))))
     (ACTIVE (PROCESS1838 SOLUTION1 SOLUTION2))
        (PRECONDITION1839 SOLUTION1 SOLUTION2)


   Selecting best theory ...


Theories are:

    ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
    ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-20
    PROCESS1797>)

    (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
    ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-15
    PROCESS1797> <CWA-24 PROCESS1838>))


Final theory is:

    ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
    ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-20
    PROCESS1797>))

Theory revision completed

NIL

>(process-definitions-from-name 'process1797)

  ((DEFPROCESS (PROCESS1797 ?VAR1807 ?VAR1808 ?VAR1809)
     INDIVIDUALS          ((?VAR1807 (CONTAINED-FLUID ?VAR1807)
                                (CONTAINED-LIQUID ?VAR1807))
                           (?VAR1808 (CONTAINED-FLUID ?VAR1808)
                                (CONTAINED-LIQUID ?VAR1808))
                           (?VAR1809 (PATH ?VAR1809)))
     PRECONDITIONS        ((PRECONDITION1806 ?VAR1807 ?VAR1808 ?VAR1809))
     QUANTITYCONDITIONS   NIL
     RELATIONS            ((Q+ (PROCESS1797-RATE ?SELF)
                                (CROSS-SECTIONAL-AREA ?VAR1809)))
     INFLUENCES           ((I+ (AMOUNT-OF (SOLVENT-OF ?VAR1808))
                                (A (PROCESS1797-RATE ?SELF)))
                           (I- (AMOUNT-OF (SOLVENT-OF ?VAR1807))
                                (A (PROCESS1797-RATE ?SELF))))))
```

A.7. Episode 4: Learning a New Quantity Condition

COAST is asked to explain why the amount of solution1 remains constant in a scenario similar to the previous scenarios except that in this scenario the concentrations of the two solutions are equal.

```
>(explain-observation '(:constant (ds (amount-of solution2))) (EVAL
*osmosis-saturation-scenario*))

Explaining (CONSTANT (DS (AMOUNT-OF SOLUTION2))) in <S-1
OSMOSIS-SATURATION-SCENARIO> with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION>
```

<CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-20 PROCESS1797>)

Theory (<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-20 PROCESS1797>) failed to explain (CONSTANT (DS (AMOUNT-OF SOLUTION2))) in <S-1 OSMOSIS-SATURATION-SCENARIO>

Explanation-based theory revision ...

Failure type: BROKEN-EXPLANATION
Failure subtype: FAILED-PREDICTION

COAST predicts that the amount of solution2 increases because the new process, Process1797, is active. This leads to a failed prediction.

Generating abstract hypotheses ...

The hypotheses are:

((<TR-H-1 (INACTIVE? (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?)
(<TR-H-3 (EQUALS? (DECREASE (DS (AMOUNT-OF SOLUTION2))) (INCREASE (DS (AMOUNT-OF SOLUTION2))))> EQUALS?)
(<TR-H-4 (NOT-CAUSES? (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH) (CONSTANT (DS (AMOUNT-OF SOLUTION2))))> NOT-CAUSES?))

Experimentation-based hypothesis refutation ...

Exemplar-based theory rejection ...

Refining hypotheses ...

The hypotheses are:

((<TR-H-16 (NEW-PRECONDITION (PRECONDITION8 SOLUTION1 SOLUTION2 PARTITION-PATH) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-15 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF (SOLVENT-OF SOLUTION2))) (A (AMOUNT-OF (SOLVENT-OF SOLUTION1)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-14 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION1)) (A (AMOUNT-OF (SOLVENT-OF SOLUTION2)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-13 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (CONCENTRATION SOLUTION2)) (A (CONCENTRATION SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-12 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION2)) (A (AMOUNT-OF (SOLVENT-OF SOLUTION1)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-11 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION2)) (A (AMOUNT-OF SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-10 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF (SOLVENT-OF SOLUTION1))) (A (AMOUNT-OF-LIMIT (SOLVENT-OF SOLUTION1)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-9 (NEW-QUANTITY-CONDITION (LESS-THAN (A (AMOUNT-OF (SOLVENT-OF

SOLUTION2))) (A (AMOUNT-OF-LIMIT (SOLVENT-OF SOLUTION2)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-8 (NEW-QUANTITY-CONDITION (LESS-THAN (A (CONCENTRATION SOLUTION1)) (A (CONCENTRATION-LIMIT SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-7 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION1)) (A (AMOUNT-OF-LIMIT SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-6 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (CONCENTRATION SOLUTION2)) (A (CONCENTRATION-LIMIT SOLUTION2))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?)
(<TR-H-5 (NEW-QUANTITY-CONDITION (LESS-THAN (A (AMOUNT-OF SOLUTION2)) (A (AMOUNT-OF-LIMIT SOLUTION2))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> INACTIVE?-CONDITIONS?))

Experimentation-based hypothesis refutation ...

A few of the experiments designed to test the hypotheses are described below.

Scenario is:
<S-13 OSMOSIS-SATURATION-SCENARIO>:
Transformations: (((LESS-THAN (A (AMOUNT-OF SOLUTION2)) (A (AMOUNT-OF-LIMIT SOLUTION2))) . TRUE))

Building ELABORATION experiment-11 for (DS (AMOUNT-OF SOLUTION2)) in <S-13 OSMOSIS-SATURATION-SCENARIO>

Showing values supported by hypotheses:
<PREDICTION-124 (DS (AMOUNT-OF SOLUTION2)) for <S-13 OSMOSIS-SATURATION-SCENARIO>
(CONSTANT <TR-H-16 (NEW-PRECONDITION (PRECONDITION8 SOLUTION1 SOLUTION2 PARTITION-PATH) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-15 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF (SOLVENT-OF SOLUTION2))) (A (AMOUNT-OF (SOLVENT-OF SOLUTION1)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-14 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION1)) (A (AMOUNT-OF (SOLVENT-OF SOLUTION2)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-13 (NEW-QUANTITY-CONDITION (GREATEP-THAN (A (CONCENTRATION SOLUTION2)) (A (CONCENTRATION SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-12 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION2)) (A (AMOUNT-OF (SOLVENT-OF SOLUTION1)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-11 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION2)) (A (AMOUNT-OF SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-10 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF (SOLVENT-OF SOLUTION1))) (A (AMOUNT-OF-LIMIT (SOLVENT-OF SOLUTION1)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-9 (NEW-QUANTITY-CONDITION (LESS-THAN (A (AMOUNT-OF (SOLVENT-OF SOLUTION2))) (A (AMOUNT-OF-LIMIT (SOLVENT-OF SOLUTION2)))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-8 (NEW-QUANTITY-CONDITION (LESS-THAN (A (CONCENTRATION SOLUTION1)) (A (CONCENTRATION-LIMIT SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-7 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF SOLUTION1)) (A (AMOUNT-OF-LIMIT SOLUTION1))) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
<TR-H-6 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (CONCENTRATION

```
        SOLUTION2)) (A (CONCENTRATION-LIMIT SOLUTION2))) (PROCESS1797 SOLUTION1
        SOLUTION2 PARTITION-PATH))>)
     (INCREASE   <TR-H-5 (NEW-QUANTITY-CONDITION (LESS-THAN (A (AMOUNT-OF
        SOLUTION2)) (A (AMOUNT-OF-LIMIT SOLUTION2))) (PROCESS1797 SOLUTION1
        SOLUTION2 PARTITION-PATH))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
    Scenario: <S-13 OSMOSIS-SATURATION-SCENARIO>

  Performing ELABORATION experiment 11 <S-13 OSMOSIS-SATURATION-SCENARIO>:
    (DS (AMOUNT-OF SOLUTION2)) = CONSTANT

  Refuting <TR-H-5 (NEW-QUANTITY-CONDITION (LESS-THAN (A (AMOUNT-OF SOLUTION2))
    (A (AMOUNT-OF-LIMIT SOLUTION2))) (PROCESS1797 SOLUTION1 SOLUTION2
    PARTITION-PATH))> based on
    <PREDICTION-124 (DS (AMOUNT-OF SOLUTION2)) for <S-13
    OSMOSIS-SATURATION-SCENARIO>
Scenario is:
  <S-5 OSMOSIS-SATURATION-SCENARIO>:
  Transformations: (((GREATER-THAN (A (CONCENTRATION SOLUTION2)) (A
  (CONCENTRATION SOLUTION1))) . TRUE))

  Building ELABORATION experiment-40 for (DS (CONCENTRATION SOLUTION1)) in <S-5
    OSMOSIS-SATURATION-SCENARIO>

  Building ELABORATION experiment-41 for (DS (AMOUNT-OF SOLUTION1)) in <S-5
    OSMOSIS-SATURATION-SCENARIO>

  Building ELABORATION experiment-42 for (DS (CONCENTRATION SOLUTION2)) in <S-5
    OSMOSIS-SATURATION-SCENARIO>

  Building ELABORATION experiment-43 for (DS (AMOUNT-OF SOLUTION2)) in <S-5
    OSMOSIS-SATURATION-SCENARIO>

  Showing values supported by hypotheses:
    <PREDICTION-476 (DS (AMOUNT-OF SOLUTION2)) for <S-5
    OSMOSIS-SATURATION-SCENARIO>
     (CONSTANT   <TR-H-16 (NEW-PRECONDITION (PRECONDITION8 SOLUTION1 SOLUTION2
        PARTITION-PATH) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
        <TR-H-15 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF
        (SOLVENT-OF SOLUTION2))) (A (AMOUNT-OF (SOLVENT-OF SOLUTION1))))
        (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))>
        <TR-H-14 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF
        SOLUTION1)) (A (AMOUNT-OF (SOLVENT-OF SOLUTION2)))) (PROCESS1797
        SOLUTION1 SOLUTION2 PARTITION-PATH))>)
     (INCREASE   <TR-H-13 (NEW-QUANTITY-CONDITION (GREATER-THAN (A
        (CONCENTRATION SOLUTION2)) (A (CONCENTRATION SOLUTION1))) (PROCESS1797
        SOLUTION1 SOLUTION2 PARTITION-PATH))>)

  ELABORATION Experiment:
    Quantity to be measured: (DS (AMOUNT-OF SOLUTION2))
    Scenario: <S-5 OSMOSIS-SATURATION-SCENARIO>

  Performing ELABORATION experiment 43 <S-5 OSMOSIS-SATURATION-SCENARIO>:
    (DS (AMOUNT-OF SOLUTION2)) = INCREASE

  Refuting <TR-H-16 (NEW-PRECONDITION (PRECONDITION8 SOLUTION1 SOLUTION2
    PARTITION-PATH) (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> based on
    <PREDICTION-476 (DS (AMOUNT-OF SOLUTION2)) for <S-5
    OSMOSIS-SATURATION-SCENARIO>

  Refuting <TR-H-15 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF
    (SOLVENT-OF SOLUTION2))) (A (AMOUNT-OF (SOLVENT-OF SOLUTION1))))
    (PROCESS1797 SOLUTION1 SOLUTION2 PARTITION-PATH))> based on
```

<PREDICTION-476 (DS (AMOUNT-OF SOLUTION2)) for <S-5
OSMOSIS-SATURATION-SCENARIO>

Refuting <TR-H-14 (NEW-QUANTITY-CONDITION (GREATER-THAN (A (AMOUNT-OF
SOLUTION1)) (A (AMOUNT-OF (SOLVENT-OF SOLUTION2)))) (PROCESS1797 SOLUTION1
SOLUTION2 PARTITION-PATH))>
based on <PREDICTION-476 (DS (AMOUNT-OF SOLUTION2)) for <S-5
OSMOSIS-SATURATION-SCENARIO>

Showing values supported by hypotheses:
<PREDICTION-477 (DS (CONCENTRATION SOLUTION2)) for <S-5
OSMOSIS-SATURATION-SCENARIO>
(CONSTANT)
(DECREASE <TR-H-13 (NEW-QUANTITY-CONDITION (GREATER-THAN (A
(CONCENTRATION SOLUTION2)) (A (CONCENTRATION SOLUTION1))) (PROCESS1797
SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
Quantity to be measured: (DS (CONCENTRATION SOLUTION2))
Scenario: <S-5 OSMOSIS-SATURATION-SCENARIO>

Performing ELABORATION experiment 42 <S-5 OSMOSIS-SATURATION-SCENARIO>:
(DS (CONCENTRATION SOLUTION2)) = DECREASE

Showing values supported by hypotheses:
<PREDICTION-478 (DS (AMOUNT-OF SOLUTION1)) for <S-5
OSMOSIS-SATURATION-SCENARIO>
(CONSTANT)
(DECREASE <TR-H-13 (NEW-QUANTITY-CONDITION (GREATER-THAN (A
(CONCENTRATION SOLUTION2)) (A (CONCENTRATION SOLUTION1))) (PROCESS1797
SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
Quantity to be measured: (DS (AMOUNT-OF SOLUTION1))
Scenario: <S-5 OSMOSIS-SATURATION-SCENARIO>

Performing ELABORATION experiment 41 <S-5 OSMOSIS-SATURATION-SCENARIO>:
(DS (AMOUNT-OF SOLUTION1)) = DECREASE

Showing values supported by hypotheses:
<PREDICTION-479 (DS (CONCENTRATION SOLUTION1)) for <S-5
OSMOSIS-SATURATION-SCENARIO>
(CONSTANT)
(INCREASE <TR-H-13 (NEW-QUANTITY-CONDITION (GREATER-THAN (A
(CONCENTRATION SOLUTION2)) (A (CONCENTRATION SOLUTION1))) (PROCESS1797
SOLUTION1 SOLUTION2 PARTITION-PATH))>)

ELABORATION Experiment:
Quantity to be measured: (DS (CONCENTRATION SOLUTION1))
Scenario: <S-5 OSMOSIS-SATURATION-SCENARIO>

Performing ELABORATION experiment 40 <S-5 OSMOSIS-SATURATION-SCENARIO>:
(DS (CONCENTRATION SOLUTION1)) = UNKNOWN

Exemplar-based theory rejection ...

Refining hypotheses ...

No more refinement ...

Explaining (CONSTANT (DS (AMOUNT-OF SOLUTION2))) in <S-1
OSMOSIS-SATURATION-SCENARIO> with theory: (<CWA-1 SOLUTION> <CWA-2 EVAPORATION>

```
<CWA-3 CONDENSATION> <CWA-4 ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE>
<CWA-7 FLUID-FLOW> <CWA-33 PROCESS1797>)

  Explanation 14695 for (CONSTANT (DS (AMOUNT-OF SOLUTION2)))
    (I- (AMOUNT-OF SOLUTION2) (A (EVAPORATION-RATE (EVAPORATION SOLUTION2 VAPOR2))))
       (INACTIVE (EVAPORATION SOLUTION2 VAPOR2))
       ...

Final theory is:

    ((<CWA-1 SOLUTION> <CWA-2 EVAPORATION> <CWA-3 CONDENSATION> <CWA-4
    ABSORPTION> <CWA-5 RELEASE> <CWA-6 ADD-SOLUTE> <CWA-7 FLUID-FLOW> <CWA-33
    PROCESS1797>))

Theory revision completed

NIL

>(process-definitions-from-name 'process1797)

  ((DEFPROCESS (PROCESS1797 ?VAR1807 ?VAR1808 ?VAR1809)
      INDIVIDUALS              ((?VAR1807 (CONTAINED-FLUID ?VAR1807)
                                      (CONTAINED-LIQUID ?VAR1807))
                                (?VAR1808 (CONTAINED-FLUID ?VAR1808)
                                      (CONTAINED-LIQUID ?VAR1808))
                                (?VAR1809 (PATH ?VAR1809)))
      PRECONDITIONS            ((PRECONDITION1806 ?VAR1807 ?VAR1808 ?VAR1809))
      QUANTITYCONDITIONS       ((GREATER-THAN (A (CONCENTRATION ?VAR1808))
                                      (A (CONCENTRATION ?VAR1807))))
      RELATIONS                ((Q+ (PROCESS1797-RATE ?SELF)
                                      (CROSS-SECTIONAL-AREA ?VAR1809)))
      INFLUENCES               ((I- (AMOUNT-OF (SOLVENT-OF ?VAR1807))
                                      (A (PROCESS1797-RATE ?SELF)))
                                (I+ (AMOUNT-OF (SOLVENT-OF ?VAR1808))
                                      (A (PROCESS1797-RATE ?SELF))))))>
```

REFERENCES

[Amarel86] S. Amarel, "Program Synthesis as a Theory Formation Task: Problem Representations and Solution Methods," In *Machine Learning: An Artificial Intelligence Approach, Vol. II*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (ed.), Los Altos, CA, 1986, pp. Morgan Kaufmann Inc..

[Bareiss87] E. R. Bareiss and B. W. Porter, "PROTOS: An Exemplar-Based Learning Apprentice," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, June 1987.

[Bennett87] S. W. Bennett, "Approximation in Mathematical Domains," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 239-241. (Also appears as Technical Report UILU-ENG-87-2238, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Buchanan84] B. G. Buchanan and E. H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA, 1984.

[Carbonell87] J. G. Carbonell and Y. Gil, "Learning by Experimentation," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, June 1987, pp. 256-266.

[Chien87] S. A. Chien, "Simplifications in Temporal Persistence: An Approach to the Intractable Domain Theory Problem in Explanation-Based Learning," M.S. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, August 1987. (Also appears as UILU-ENG-87-2255, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Danyluk87] A. P. Danyluk, "The Use of Explanations for Similarity-Based Learning," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 274-276.

[Davis84] R. Davis, "Diagnostic Reasoning based on Structure and Behavior," *Artificial Intelligence 24*, (1984)..

[DeJong86] G. F. DeJong and R. J. Mooney, "Explanation-Based Learning: An Alternative View," *Machine Learning 1*, 2 (April 1986), pp. 145-176. (Also appears as Technical Report UILU-ENG-86-2208, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[de Kleer84] J. de Kleer and J. S. Brown, "A Qualitative Physics Based on Confluences," *Artificial Intelligence 24*, (1984)..

[de Kleer87] J. de Kleer and B. C. Williams, "Diagnosing Multiple Faults," *Artificial Intelligence 32*, 1 (1987),.

[Dietterich83] T G. Dietterich and R. S. Michalski, "A Comparative Review of Selected Methods for Learning from Examples," In *Machine Learning: An Artificial Intelligence Approach*, R S. Michalski, J G Carbonell and T M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, pp. 41–81.

[Dietterich86a] T. G. Dietterich, "Learning at the Knowledge Level," *Machine Learning 1*, 3 (1986), pp. 287–316.

[Dietterich86b] T. G. Dietterich, "The EG Project: Recent Progress," In *Machine Learning: A Guide To Current Research*, T. M. Mitchell, J. G. Carbonell and R. S. Michalski (ed.), Kluwer Academic Publishers, Hingham, MA, 1986, pp. 51–54

[Dietterich88] T. Dietterich and N. Flann, "An Inductive Approach to Solving the Imperfect Theory Problem," *Proceedings of the 1988 AAAI Spring Symposium Series on Explanation-based Learning*, Stanford, CA, March 1988.

[Doyle86] R. J. Doyle, "Constructing and Refining Causal Explanations from an Inconsistent Domain Theory," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 538–544.

[Ellman85] T. Ellman, "Generalizing Logic Circuit Designs by Analyzing Proofs of Correctness," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 643–646.

[Ellman88] T. Ellman, "Approximate Theory Formation: An Explanation-based Approach," *Proceedings of the Seventh National Conference on Artificial Intelligence*, Saint Paul, MN, August 1988.

[Falkenhainer86] B. C. Falkenhainer and R. S. Michalski, "Integrating Quantitative and Qualitative Discovery: The ABACUS System," *Machine Learning 1*, 4 (1986), pp. 367–401.

[Falkenhainer87a] B. Falkenhainer, "Scientific Theory Formation Through Analogical Inference," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, June 1987, pp. 218–229.

[Falkenhainer87b] B. Falkenhainer, "An Examination of the Third Stage in the Analogy Process: Verification-based Analogical Learning," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987.

[Falkenhainer88a] B. Falkenhainer, "The Utility of Difference-Based Reasoning," *Proceedings of the Seventh National Conference on Artificial Intelligence*, Saint Paul, MN, August 1988.

[Falkenhainer88b] B. Falkenhainer and S. A. Rajamoney, "The Interdependencies of Theory Formation, Revision, and Experimentation," *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, MI, June 1988.

[Falkenhainer89] B. Falkenhainer, "Learning from Physical Analogies: A Study in Analogy and the Explanation Process," Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, January, 1989.

[Forbus83] K. Forbus and D. Gentner, "Learning Physical Domains: Towards a Theoretical Framework," *Proceedings of the 1983 International Machine Learning Workshop*, Urbana, IL, June 1983, pp. 198–202.

[Forbus84a] K. D. Forbus, "Qualitative Process Theory," *Artificial Intelligence 24*, (1984), pp. 85–168.

[Forbus84b] K. D. Forbus, "Qualitative Process Theory," Technical Report 789, Ph.D. Thesis, MIT AI Lab, Cambridge, MA, August 1984.

[Genesereth84] M. R. Genesereth, "The Use of Design Descriptions in Automated Diagnosis," *Artificial Intelligence 24*, (1984),.

[Gentner83] D. Gentner, "Structure-Mapping: A Theoretical Framework for Analogy," *Cognitive Science 7*, (1983), pp. 155–170.

[Ginsberg88] A. Ginsberg, "Theory Revision via Prior Operationalization," *Proceedings of the Seventh National Conference on Artificial Intelligence*, Saint Paul, MN, August 1988.

[Hall86] R. J. Hall, "Learning by Failing to Explain," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986.

[Hammond86] K. Hammond, "CHEF: A Model of Case-Based Planning," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 267–271.

[Hirsh87] H. Hirsh, "Explanation-Based Generalization in a Logic-Programming Environment," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 221–227.

[Jones86] R. Jones, "Generating Predictions to Aid the Scientific Discovery Process," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 513–517.

[Kedar-Cabelli87] S. T. Kedar-Cabelli and L. T. McCarty, "Explanation-Based Generalization as Resolution Theorem Proving," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, June 1987, pp. 383–389.

[Kodratoff87a] Y. Kodratoff and G. Tecuci, "DISCIPLE-I: Interactive Apprentice System in Weak Theory Fields," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987.

[Kodratoff87b] Y. Kodratoff and G. Tecuci, "What is an Explanation in DISCIPLE?," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, June 1987.

[Kolodner87] J. L. Kolodner, "Extending Problem Solver Capabilities Through Case-Based Inference," *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, June 1987, pp. 167–178.

[Kuhn70] T. S. Kuhn, *The Structure of Scientific Revolutions, 2nd ed.*, University of Chicago Press, Chicago, IL, 1970.

[Kuipers84] B. Kuipers, "Commonsense Reasoning About Causality: Deriving Behavior from Structure," *Artificial Intelligence 24*, (1984), pp. 169–204.

[Laird87] J. E. Laird, A. Newell and P. S. Rosenbloom, "SOAR: An Architecture for General Intelligence," *Artificial Intelligence 33*, 1 (1987), pp. 1–64.

[Laird88] J. Laird, "Recovery from Incorrect Knowledge in SOAR," *Proceedings of the Seventh National Conference on Artificial Intelligence*, Saint Paul, MN, August 1988.

[Langley86] P. Langley, J. M. Zytkow, H. A. Simon and G. L. Bradshaw, "The Search for Regularity: Four Aspects of Scientific Discovery,"In *Machine Learning: An Artificial Intelligence Approach, Vol. II*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (ed.), Morgan Kaufmann, Los Altos, CA, 1986, pp. 425–469.

[Lebowitz86a] M. Lebowitz, "Not the Path to Perdition: The Utility of Similarity-Based Learning," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 533–537.

[Lebowitz86b] M. Lebowitz, "Integrated Learning: Controlling Explanation," *Cognitive Science 10*, 2 (1986), pp. 219–240.

[Lenat83] D. B. Lenat, "The Role of Heuristics in Learning by Discovery: Three Case Studies,"in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, pp. 243–306.

[Mahadevan85] S. Mahadevan, "Verification-Based Learning: A Generalization Strategy for Inferring Problem-Reduction Methods," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 616–623.

[Michalski83a] R. S. Michalski, "A Theory and Methodology of Inductive Learning,"In *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, pp. 83–134.

[Michalski83b] R. S. Michalski and R. E. Stepp, "Learning from Observation: Conceptual Clustering,"In *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, pp. 331–363.

[Miller82] R. Miller, H. Pople and J. Myers, "Internist-1, an Experimental Computer-Based Diagnostic Consultant for General Internal Medicine," *New England Journal of Medicine 307*, (1982), pp. 468–476.

[Minton84] S. N. Minton, "Constraint-Based Generalization: Learning Game-Playing Plans from Single Examples," *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, August 1984, pp. 251-254.

[Minton87] S. Minton and J. G. Carbonell, "Strategies for Learning Search Control Rules: An Explanation-based Approach," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 228-235.

[Mitchell78] T. M. Mitchell, "Version Spaces: An Approach to Concept Learning," Ph.D. Thesis, Stanford University, Palo Alto, CA, 1978. (Also appears as Technical Report STAN-CS-78-711, Stanford University)

[Mitchell85] T. M. Mitchell, S. Mahadevan and L. I. Steinberg, "LEAP: A Learning Apprentice for VLSI Design," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 573-580.

[Mitchell86] T. M. Mitchell, R. Keller and S. Kedar-Cabelli, "Explanation-Based Generalization: A Unifying View," *Machine Learning 1*, 1 (January 1986), pp. 47-80.

[Mooney86] R. J. Mooney and S. W. Bennett, "A Domain Independent Explanation-Based Generalizer," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 551-555. (Also appears as Technical Report UILU-ENG-86-2216, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Mooney88] R. J. Mooney, "A General Explanation-Based Learning Mechanism and its Application to Narrative Understanding," Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, January 1988. (Also appears as UILU-ENG-87-2269, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Mostow87] J. Mostow and T. Fawcett, "Approximating Intractable Theories: A Problem Space Model," Machine Learning-Technical Report-16, Department of Computer Science, Rutgers University, New Brunswick, New Jersey, December, 1987.

[Nordhausen87] B. Nordhausen and P. Langley, "Towards an Integrated Discovery System," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987.

[O'Rorke87] P. V. O'Rorke, "Explanation-Based Learning Via Constraint Posting and Propagation," Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, January 1987. (Also appears as UILU-ENG-87-2239, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Pazzani88a] M. J. Pazzani, "Integrated Learning with Incorrect and Incomplete Theories," *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, MI, June 1988.

[Pazzani88b] M. J. Pazzani, "Selecting the Best Explanation for Explanation-based Learning," *Proceedings of the 1988 AAAI Spring Symposium Series on Explanation-based Learning*, Stanford, CA, March 1988.

[Popper68] K. Popper, *The Logic of Scientific Discovery*, Harper and Row, New York, 1968.

[Quinlan83] J. R. Quinlan, "Learning Efficient Classification Procedures and their Application to Chess End Games," In *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA, 1983.

[Raiman86] O. Raiman, "Order of Magnitude Reasoning," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 100–104.

[Rajamoney85] S. Rajamoney, G. F. DeJong and B. Faltings, "Towards a Model of Conceptual Knowledge Acquisition through Directed Experimentation," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985. (Also appears as Working Paper 68, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Rajamoney86a] S. A. Rajamoney, "Automated Design of Experiments for Refining Theories," M. S. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, May 1986. (Also appears as Technical Report UILU-ENG-86-2213, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Rajamoney86b] S. A. Rajamoney, "Conceptual Knowledge Acquisition through Directed Experimentation," In *Machine Learning: A Guide To Current Research*, T. M. Mitchell, J. G. Carbonell and R. S. Michalski (ed.), Kluwer Academic Publishers, Hingham, MA, 1986, pp. 255–260.

[Rajamoney87] S. Rajamoney and G. DeJong, "The Classification, Detection and Handling of Imperfect Theory Problems," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 205–207. (Also appears as Technical Report UILU-ENG-87-2224, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Rajamoney88a] S. A. Rajamoney, "Experimentation-based Theory Revision," *Proceedings of the 1988 AAAI Spring Symposium Series on Explanation-based Learning*, Stanford, CA, March 1988.

[Rajamoney88b] S. A. Rajamoney and G. F. DeJong, "Active Explanation Reduction: An Approach to the Multiple Explanations Problem," *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, MI, June 1988.

[Reiter87] R. Reiter, "A Theory of Diagnosis from First Principles," *Artificial Intelligence 32*, 1 (1987),.

[Rose86] D. Rose and P. Langley, "STAHLp: Belief Revision in Scientific Discovery," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 528–532.

[Rosenbloom86] P. Rosenbloom and J. Laird, "Mapping Explanation-Based Generalization into Soar," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 561–567.

[Sacerdoti74] E. Sacerdoti, "Planning in a Hierarchy of Abstraction Spaces," *Artificial Intelligence 5*, (1974), pp. 115–135.

[Sacerdoti77] E. Sacerdoti, *A Structure for Plans and Behavior*, American Elsevier, New York, 1977.

[Segre87] A. M. Segre, "Explanation-Based Learning of Generalized Robot Assembly Tasks," Ph.D. Thesis, Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL, January 1987. (Also appears as UILU-ENG-87-2208, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Shavlik85] J. W. Shavlik, "Learning about Momentum Conservation," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 667–669. (Also appears as Working Paper 66, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Shrager87] J. Shrager, "Theory Change Via View Application in Instructionless Learning," *Machine Learning 2*, 3 (1987),.

[Sleeman82] D. H. Sleeman and J. S. Brown (ed.), *Intelligent Tutoring Systems*, Academic Press, New York, NY, 1982.

[Stefik81] M. Stefik, "Planning with Constraints (MOLGEN: Part 1)," *Artificial Intelligence 16*, 2 (1981), pp. 111–140.

[Stepp86] R. E. Stepp and R. S. Michalski, "Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects," In *Machine Learning: An Artificial Intelligence Approach, Vol. II*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (ed.), Morgan Kaufmann, Los Altos, CA, 1986, pp. 471–498.

[Thagard85] P. Thagard and K. Holyoak, "Discovering the Wave Theory of Sound: Inductive Inference in the Context of Problem Solving," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985.

[Thagard88] P. Thagard, "The Conceptual Structure of the Chemical Revolution," CSL Report 27, Cognitive Science Laboratory, Princeton University, Princeton, NJ, June, 1988.

[Weld87] D. Weld, "Comparative Analysis," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987.

[Wenger87] E. Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann, Los Altos, CA, 1987.

[Williams84] B. Williams, "Qualitative Analysis of MOS Circuits," *Artificial Intelligence 24*, (1984),.

[Winston83] P. H. Winston, T. O. Binford, B. Katz and M. Lowry, "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents," *Proceedings of the National Conference on Artificial Intelligence*, Washington, D.C., August 1983, pp. 433–439.

VITA

Shankar Anandsubramaniam Rajamoney [redacted]. He attended the Indian Institute of Technology, Madras, where he received a B. Tech. degree in Electrical Engineering (Electronics) in 1983. In 1983, he was awarded a University of Illinois fellowship. He joined the University of Illinois at Urbana–Champaign in the fall of 1983 for graduate study in Computer Science. In the summer of 1984, he became a research assistant to Professor DeJong in the Artificial Intelligence Research Group of the Coordinated Science Laboratory. He received a M.S. degree in Computer Science in 1986. He was awarded University of Illinois Artificial Intelligence/Cognitive Science fellowships from the summer of 1986 to the summer of 1988. In January 1989, he received a Ph.D. in Computer Science from the University of Illinois at Urbana–Champaign.

Dr. Rajamoney's research interests include machine learning, qualitative reasoning, robotics, and cognitive science. He has presented several papers at scientific conferences on Artificial Intelligence.

Dr. Rajamoney is currently an Assistant Professor in the Computer Science Department of the University of Southern California at Los Angeles.